BEST AVAILABLE COPY

REMARKS

Turning to the art rejections, claims 1-12 have been cancelled and replaced with new claims 13-24 to obviate the informalities objections. Claims 13-24 are identical to original claims 1-12 albeit with the spacing corrected and claim dependencies updated.

With regard to cipher 9 of the Action, a certified copy of Applicant's priority application was filed on May 1, 2002. A copy of the return acknowledgment postcard is enclosed indicating same was received by the Patent Office. Therefore, Applicant respectfully requests correction of the record.

Turning to the art rejections, claims 1 and 7 have been rejected as being obvious over Ishiwaki (U.S. Patent 6,725,415) in view of Hyodo et al. (U.S. Patent 5,282,215); claims 2 and 8 have been rejected as being obvious over Ishiwaki in view of Hyodo et al. and further in view of Henriksen (U.S. Patent 6,324,670); claims 3, 5, 9, and 11 have been rejected as being obvious over Ishiwaki in view of Hyodo et al. and further in view of Douady et al. (U.S. Patent 6,516,004); and claims 4, 6, 10, and 12 have been rejected as being obvious over Ishiwaki in view of Hyodo et al. and Douady et al. and Henriksen. All of these rejections use Ishiwaki as the primary reference.

Enclosed herewith is a Declaration under 37 CFR 1.131 which establishes that the Applicant completed the claimed invention before the U.S. filing date of the Ishiwaki reference. In this regard, Applicant notes that the Ishiwaki Patent, which issued after the filing date of the subject Application, has a January 26, 2001 U.S. filing date. Since this Application was filed on or after January 1, 1996, pursuant to § 531(b) of Public Law 10-465, the Amendment made to 35 USC § 104 is applicable in this case. Accordingly, Applicant's Declaration under 37
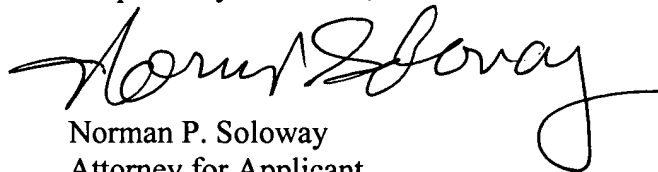
CFR 1.131, which clearly shows completion of the claimed invention in a WTO member

country, i.e., Japan, prior to the January 26, 2001 filing date of the Ishiwaki Patent removes the

Ishiwaki Patent as citable prior art under 35 USC § 102/103. Accordingly, the several

rejections of the claims as obvious from Ishiwaki in view of in view of Hyodo et al., Douady et

al. and Henriksen, cannot be maintained.

Having dealt with all the objections raised by the Examiner, the Application is believed

to be in order for allowance. Early and favorable action are respectfully requested.

In the event there are any fee deficiencies or additional fees are payable, please charge

them (or credit any overpayment) to our Deposit Account Number 08-1391.

Respectfully submitted,

Norman P. Soloway
Attorney for Applicant
Reg. No. 24,315

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States
Postal Service as First Class Mail in an envelope addressed to: MAIL STOP AMENDMENTS,
Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on
_____January 24, 2005_____, at Tucson, Arizona.

By_____Kim Hood_____

NPS/WPO:lv/sb

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Appln. Of: KOTAKA

Serial No.:   10/090,302

Filed: March 4, 2002

For: ARITHMETIC OPERATION METHOD FOR CYCLIC . . .

Group: 2133

Examiner: Dipakkumar Gandhi   DOCKET: NEC N01321

MAIL STOP AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

VERIFICATION OF TRANSLATION

Dear Sir:

The undersigned hereby certifies that I am conversant in both Japanese and English languages, that I have prepared the attached English translation of the Japanese text attached as Exhibit A, and that the English translation is a true, faithful and accurate translation of the attached Exhibit A.

I further declare that all statements made of my own knowledge are true and that all statements made on information and belief are with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC § 1001, and that such false statements may jeopardize the validity of the application or any patent issuing therefrom.

Date: January 24, 2005   Signature: _____

    Typed Name_____ *Yukuo NISHIMURA* _____

Address: ___ Nakamura The First Bldg. 3F. 4-2-6, Kitaurawa, Urawa-ku, Saitama, ___

    · SAITAMA, Japan

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

TO: Nishimura International Patent Office
Mr. Yukoo NISHIMURA

Date December 11, 2000

配布先 西村国際特許事務所 西村　征生　殿 参号：（1頁のみ） 10-22440 知財部　西野マネージャー殿	コンカレント出願 明細書作成依頼書 Request for Preparing Specification	EDC知財－特出－ 1330
		平成 12年 12月 11日
		日本電気株式会社 エレクトロンデバイス知的財産部 マネージャー 和田　　正 FROM NEC Corporation
		担当：久野　淑己
		電話：044-435-1421
		FAX：044-435-1871

下記の件について明細書作成を依頼します。

―――　記　―――

1. 整理番号　　：754-10092

2. 提案名称　　：ＣＲＣ演算回路　CRC arithmetic Operation Circuit
Title　　概要：infini bandのＣＲＣ演算においてレイテンシを削減するような回路

Inventor
3. 提案者名　　：小高　重成　shigenari KOTAKA
（三シス ・一設G）

4. 当部担当者：久野　　淑己

5. 添付書類　　：アイデア提案書
打ち合わせ記録
特開平
特開平

15日（金）
1:00～

6. 希望納期　　：草稿―　平成12年01月13日
出願―　平成12年01月20日

7. 打合せ場所：玉川　地区を予定。

8. 備考：

（第4版：1998.02.25）　本文第1／6頁

IDEA PROPOSAL	Company Reference Number	: 754-10092

Proposed Date: November 20, 2000	グループコード：	部内番号：

Authorization	Manager	Chief	主任：	Semiconductor Engineering Department Received Date	知財部 12.11.30

Inventor Entry　　　　E-mail: kotaka@lsi.nec.co.jp
Section: Third System L, First Design　TEL:822-26011　メール番号:22-26010
Name　Shigenari KOTAKA　　Company Code :0690257
適用・応用分野：通信
適用製品名：　SUN Wings　　　売上規模：　3K　　（百万円／年）
実験・試作状況：○実験・試作完了　●実験・試作中　○実験・試作予定あり　○実験・試作予定なし
Prior art search (Patent): done (JP Appln. No.H02-119319)
Prior art search (Document): done, none
特許検索式：(CRC生成回路＋CRC演算回路)＆(高速＋高速処理＋レイテンシ＋レイテンシー＋Latency)
関連提案・特許：無し　　　　　　　　　　　　1月5点
サンプル出荷／社外発表予定：○無　●有(早い方の日：H13年 9 月　　日、何処で　　　　　　)
出願希望種別：●コンカレント　○S級　○通常出願(届出予定日：1999年12月17日)

Chief Entry
上司氏名：　堀 口 立 二

Decision of reduction to practice
Planning of Foreign Application: US　　　　　　　　　　　　　　)

コメント：

[発明相談コメント欄]　センター担当：　　　　　　年　　月　　日

[評価委員会記入欄]
評価責任者氏名：　古賀隆俊
決定日：　2000年 11 月 27 日
評価結果①出願希望(●コンカレント　○S級　○通常届出) 2.公開技報 3.中止 4.再検討
外国出願希望：○無　●有(国名：●米　○韓　○中国　○台湾　○英　○独　その
他　　　　　　　　)
届出指定日：2000年 12 月 17 日
コメント：　計算方法を明確に．米国出願でのアルゴリズムの権利化を検討して
　　　　　下さい．有数バイトでの処理例を追加し権利範囲を広げて下さい．
　　　　　　　　　古賀
センターへの要望：

Chief Signature
I read on pages 1 to 6 of this proposal and have understood
Name: Tatsuji HORIGUCHI　　November 20,2000　　this invention.

Inventor signature
Name Shigenari KOTAKA　　November 20,2000

氏名：　　　　　　　20　年　　月　　日

1

[CRC]

A CRC (cyclic redundancy check) is one of methods for checking whether or not data have correctly been transmitted (read or written) in data transmission, and in writing or reading data into/from a disk, a tape, or a like.

A CRC arithmetic operation is performed by using an expression made up of a combination of shift and addition, called a CRC generative polynomial. A value generally used in the CRC arithmetic operation is made up 16 bits or 32 bits (The word "cyclic" in CRC is derived from a calculation method in which power of 2 is used as a modulus, and an over-flow of an operational result is neglected).

Since the CRC arithmetic operation is not to perform a simple addition operation, there is demerit in that processing using a software for the CRC calculation greatly increases a processor workload. On the other hand, since processing by a hardware is simple and easy, the CRC arithmetic operation method is generally used in a disk controller, a communication LSI, and a like.

Source: ASCII digital term dictionary

[CRC generative polynomial]

The above-described CRC generative polynomials are defined as follows:

CRC32(32bit) : $G(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{13}+X^{11}+X^{8}+X^{7}+X^{5}+X^{4}+X^{3}+X^{1}+1$

CRC16(16bit) : $G(X) = X^{16}+X^{12}+X^{5}+X^{1}+1$

[CRC method]

Next, the CRC arithmetic operation method will briefly be explained. The following, for simplicity's sake, example using 6 bits will be explained: because using 32 bits or 16 bits as an example makes the explanation (operational expressions) too complicated.

Please note that an arithmetic operation method using 32 bits or 16 bits is a same as that using 6 bits.

① A polynomial is given below, in which input data is considered to specify a value.

2

$$\boxed{1}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{1}$$
$$X^0 \quad X^1 \quad X^2 \quad X^3 \quad X^4 \quad X^5$$
$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$
$$P(X) = \quad 1 + X^1 + 0 + 0 + X^4 + X^5$$

② Next, the CRC generative polynomial predetermined in a data transmitting/receiving is used.
(CRC32 and CRC16 generative polynomials are shown above.)

$$CRC6\,(6bit) : G(X) = X^6 + X^3 + 1$$

③ A result obtained by multiplying the input data P(X) by the highest order term X^6 included in the generative polynomial G(X) is represented by Q(X).

$$Q(X) = X^{11} + X^{10} + X^7 + X^6$$

④ Then, the Q(X) is divided by the generative polynomial G(X) and its remainder is used as a cyclic check bit of the CRC arithmetic operation method, which is called a "CRC code".

$$
\begin{array}{r}
X^5 + X^4 + X^2 + 1 \\
X^6 + X^3 + 1 \; \overline{)\; X^{11} + X^{10} \quad\quad + X^7 + X^6} \\
X^{11} \quad\quad + X^8 \quad\quad\quad + X^5 \\
\hline
X^{10} + X^8 + X^7 + X^6 + X^5 \\
X^{10} \quad\quad + X^7 \quad\quad\quad + X^4 \\
\hline
X^8 \quad\quad + X^6 + X^5 + X^4 \\
X^8 \quad\quad\quad + X^5 \quad\quad + X^2 \\
\hline
X^6 \quad\quad + X^4 \quad\quad + X^2 \\
X^6 \quad\quad\quad\quad + X^3 \quad + 1 \\
\hline
\end{array}
$$

Remainder··· $X^4 + X^3 + X^2 + 1$

$$\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{0}$$
$$X^0 \quad X^1 \quad X^2 \quad X^3 \quad X^4 \quad X^5$$

⑤ A new Q(X) is produced by multiplying input data to be input next by the CRC code obtained by the CRC arithmetic operation ④. By dividing the new Q(X) by the generative polynomial G(X), a new CRC code is obtained.

When the processing described above is performed repeatedly (in a cyclic manner) on all the input data, a CRC arithmetic operation result can be obtained, and the obtained CRC arithmetic operation result is transmitted by being added to an end of data to be transmitted.

3

[CRC operational expression]

The concept of the CRC arithmetic operation is as described above.

However, in a multi-bit CRC32 arithmetic operation such as the CRC32 arithmetic operation, the above-described division cannot be done simply by hardware because the hardware cannot perform high-speed processing or because large-sized circuits have to be used as the hardware and, therefore, the division is generally done using the following method. (Again, an example using 6 bits will be described.)

First, the following circuit can be obtained in accordance with the above-described CRC generative polynomial.

⊗ : Exclusive-OR
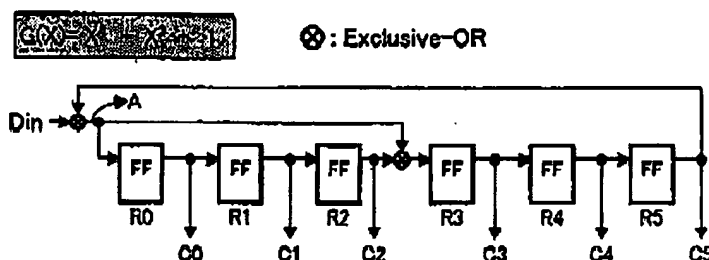
Next, when input data P(X) described earlier under the section "CRC method" is serially input through Din, an output state in each FF (flip-flop) will be given below:

-Input data-

1	1	0	0	1	1
X^0	X^1	X^2	X^3	X^4	X^5

Here, "A" denotes an Ex-OR of loop-back data and input data.

Shifted	Input value	A	FF output						Note
			C0	C1	C2	C3	C4	C5	
0	—	—	0	0	0	0	0	0	Initial value
1	1	1	1	0	0	1	0	0	
2	1	1	1	1	0	1	1	0	
3	0	0	0	1	1	0	1	1	
4	0	1	1	0	1	0	0	1	
5	1	0	0	1	0	1	0	0	
6	1	1	1	0	1	1	1	0	Remainder

4

Agreement can be seen between the remainder described in the above table and the remainder explained earlier under the section "CRC method", that is, a set of output data being output from each of the flip-flops when data are shifted sequentially by the number of its bits which is equal to that of the input data is the "CRC code" to be acquired.

The following operational expression can be introduced, wherein input data are expressed sequentially by D0-D5, each initial value of FFs is expressed by R0-R5, and each output of the FFs is expressed by C0-C5.

Shifted	Input Value	A	FF output	
0	—	—	C5	R5
			C4	R4
			C3	R3
			C2	R2
			C1	R1
			C0	R0
1	D5	R5·D5	C5	R4
			C4	R3
			C3	R2·R5·D5
			C2	R1
			C1	R0
			C0	R5·D5
2	D4	R4·D4	C5	R3
			C4	R2·R5·D5
			C3	R1·R4·D4
			C2	R0
			C1	R5·D5
			C0	R4·D4
3	D3	R3·D3	C5	R2·R5·D5
			C4	R1·R4·D4
			C3	R0·R3·D3
			C2	R5·D5
			C1	R4·D4
			C0	R3·D3
4	D2	R2·R5·D5·D2	C5	R1·R4·D4
			C4	R0·R3·D3
			C3	R5·D5·R2·R5·D5·D2
			C2	R4·D4
			C1	R3·D3
			C0	R2·R5·D5·D2

Note: "." indicates EX-OR

5

Shifted	Input value	A	FF output	
5	D1	R1·R4·D4·D1	C5	R0·R3·D3
			C4	R5·D5·R2·R5·D5·D2
			C3	R4·D4·R1·R4·D4·D1
			C2	R3·D3
			C1	R2·R5·D5·D2
			C0	R1·R4·D4·D1
6	D0	R0·R3·D3·D0	C5	R5·D5·R2·R5·D5·D2
			C4	R4·D4·R1·R4·D4·D1
			C3	R3·D3·R0·R3·D3·D0
			C2	R2·R5·D5·D2
			C1	R1·R4·D4·D1
			C0	R0·R3·D3·D0

Note: "."indicates EX-OR

Operational expressions obtained by each of output data from the flip-flops, when data shift operations are performed six times (that is, six shifts), are CRC6 operational expressions. Here, since same terms (R3 · R3 etc.) can be deleted, the following operational expressions are what to acquired, by rearranging each of the obtained operational expressions.

CRC6 operational expressions
$$C5=R2·D2$$
$$C4=R1·D1$$
$$C3=R0·D0$$
$$C2=R2·R5·D2·D5$$
$$C1=R1·R4·D1·D4$$
$$C0=R0·R3·D0·D3$$

The following expressions are obtained, by inputting initial values "0" (R0-R5="0") and data (11011) to the above-mentioned operational expressions.
$$C5=0$$
$$C4=1$$
$$C3=1$$
$$C2=1$$
$$C1=0$$
$$C0=1$$

Thus, agreement can be seen between this result and the earlier result. Accordingly, it is confirmed that the operational expressions described above are effective to use for checking error.

The above is all of the method explanation and operational expression introduction in CRC6.

6

The operational expressions of CRC16 and CRC32 can also be introduced by using a same manner as described above in CRC6.

The generative polynomials, generating circuits and operational expressions of CRC16 and CRC32 will be shown below.
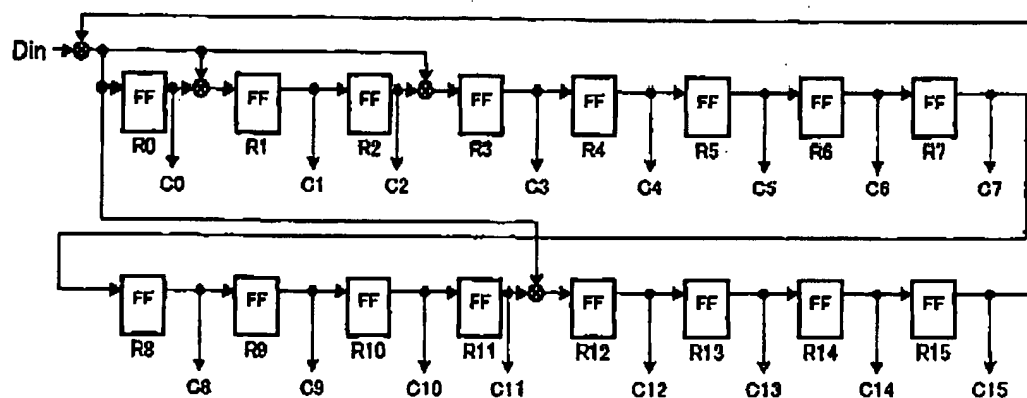
【CRC16】

Generative polynomial

$G(X) = X^{16}+X^{12}+X^3+X^1+1$

Generating circuit

$\otimes$: Exclusive-OR

Operational expressions

C0	R0·R4·R8·R12·R13·R15·D0·D2·D3·D7·D11·D15
C1	R0·R1·R4·R5·R8·R9·R12·R14·R15·D0·D1·D3·D6·D7·D10·D11·D14·D15
C2	R1·R2·R5·R6·R9·R10·R13·R15·D0·D2·D5·D6·D9·D10·D13·D14
C3	R0·R2·R3·R4·R6·R7·R8·R10·R11·R12·R13·R14·R15·D0·D1·D2·D3·D4·D5·D7·D8·D9·D11·D12·D13·D15
C4	R1·R3·R4·R5·R7·R8·R9·R11·R12·R13·R14·R15·D0·D1·D2·D3·D4·D6·D7·D8·D10·D11·D12·D14
C5	R2·R4·R5·R6·R8·R9·R10·R12·R13·R14·R15·D0·D1·D2·D3·D5·D6·D7·D9·D10·D11·D13
C6	R3·R5·R6·R7·R9·R10·R11·R13·R14·R15·D0·D1·D2·D4·D5·D6·D8·D9·D10·D12
C7	R4·R6·R7·R8·R10·R11·R12·R14·R15·D0·D1·D3·D4·D5·D7·D8·D9·D11
C8	R5·R7·R8·R9·R11·R12·R13·R15·D0·D2·D3·D4·D6·D7·D8·D10
C9	R6·R8·R9·R10·R12·R13·R14·D1·D2·D3·D5·D6·D7·D9
C10	R7·R9·R10·R11·R13·R14·R15·D0·D1·D2·D4·D5·D6·D8
C11	R8·R10·R11·R12·R14·R15·D0·D1·D3·D4·D5·D7
C12	R0·R4·R8·R9·R11·D4·D6·D7·D11·D15
C13	R1·R5·R9·R10·R12·D3·D5·D6·D10·D14
C14	R2·R6·R10·R11·R13·D2·D4·D5·D9·D13
C15	R3·R7·R11·R12·R14·D1·D3·D4·D08·D12

7

The above shows operational expressions in which the input data is made up of 16 bits (D0-D15). If input data length is different from 16 bits, other operational expressions different from the above ones have to be used. For example, in a case where the input data is made up of 8 bits (1 byte: D0-D07), an output data from each flip-flop at a time point when D7 has been input (that is, eight shifts) becomes each of the required operational expressions.
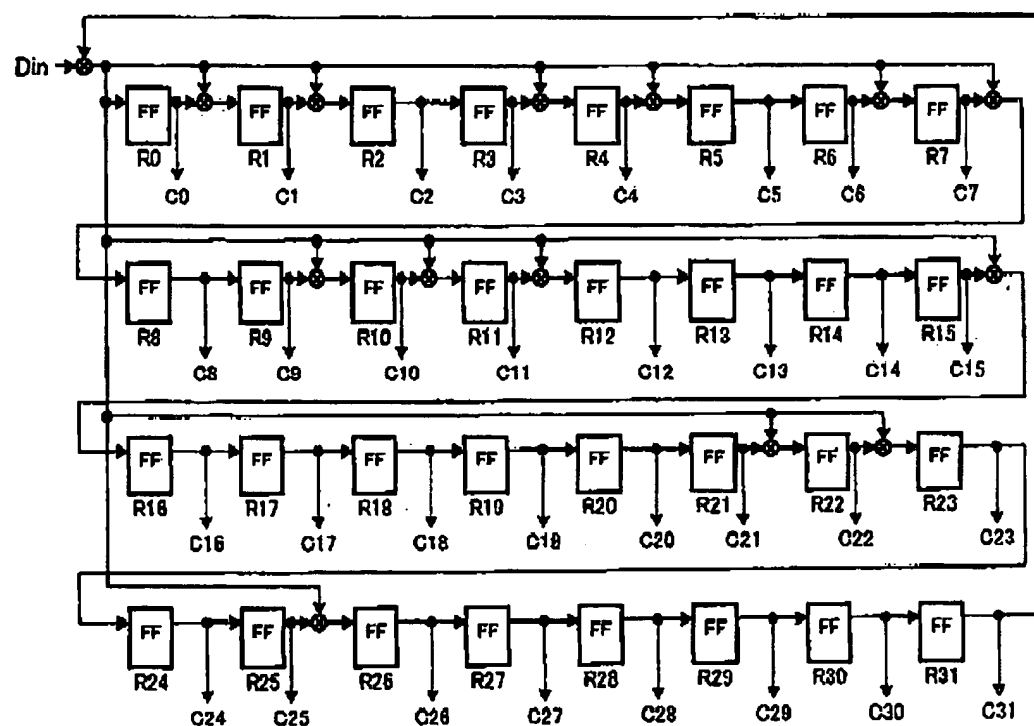
【CRC32】

Generative polynomial

CRC32(32bit) : $G(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1$

Generating circuit

⊗ : Exclusive-OR

Operational expressions

C0	R0·R6·R9·R10·R12·R16·R24·R25·R26·R28·R29·R30·R31·D0·D1·D2·D3·D5·D6·D7·D15·D19·D21·D22·D25·D31
C1	R0·R1·R6·R7·R9·R11·R12·R13·R16·R17·R24·R27·R28·D3·D4·D7·D14·D15·D18·D19·D20·D22·D24·D25·D30·D31
C2	R0·R2·R6·R7·R8·R9·R13·R14·R16·R17·R18·R24·R26·R30·R31·D0·D1·D5·D7·D13·D14·D15·D17·D18·D22·D23·D24·D25·D29·D30·D31
C3	R1·R2·R3·R7·R8·R9·R10·R14·R15·R17·R18·R19·R25·R27·R31·D0·D4·D6·D12·D13·D14·D16·D17·D21·D22·D23·D24·D28·D29·D30
C4	R0·R2·R3·R4·R6·R8·R11·R12·R15·R18·R19·R20·R24·R25·R29·R30·R31·D0·D1·D2·D6·D7·D11·D12·D13·D16·D19·D20·D23·D25·D27·D28·D29·D31
C5	R0·R1·R3·R4·R5·R6·R7·R10·R13·R19·R20·R21·R24·R28·R29·D2·D3·D7·D10·D11·D12·D18·D21·D24·D25·D26·D27·D28·D30·D31
C6	R1·R2·R4·R5·R6·R7·R8·R11·R20·R21·R25·R30·D1·D2·D6·D9·D10·D11·D17·D20·D23·D24·D25·D26·D27·D29·D30
C7	R0·R2·R3·R5·R7·R8·R10·R15·R16·R21·R22·R23·R24·R28·R29·D2·D3·D6·D7·D8·D9·D10·D15·D16·D21·D23·D24·D26·D28·D29·D31
C8	R0·R1·R3·R4·R8·R10·R11·R17·R22·R28·R31·D0·D3·D8·D9·D14·D19·D20·D21·D23·D27·D28·D30·D31
C9	R1·R2·R4·R5·R9·R11·R12·R13·R18·R23·R24·R29·D2·D7·D8·D13·D18·D19·D20·D22·D26·D27·D29·D30
C10	R0·R2·R3·R5·R9·R13·R14·R16·R19·R26·R28·R29·R31·D0·D2·D3·D5·D12·D15·D17·D18·D22·D26·D28·D29·D31
C11	R0·R1·R3·R4·R9·R12·R14·R15·R16·R17·R20·R24·R25·R26·R27·R28·R31·D0·D3·D4·D5·D6·D7·D11·D14·D15·D16·D17·D19·D22·D27·D28·D30·D31
C12	R0·R1·R2·R4·R5·R6·R9·R12·R13·R15·R17·R18·R24·R30·R31·D0·D1·D4·D7·D10·D13·D14·D16·D18·D19·D22·D25·D26·D27·D29·D30·D31
C13	R1·R2·R3·R5·R6·R7·R10·R13·R16·R19·R22·R28·R31·D0·D3·D6·D9·D12·D13·D15·D17·D18·D21·D24·D25·D26·D28·D29·D30
C14	R02 R03 R04 R06 R07 R08 R11 R14 R15 R17 R19 R20 R23 R26 R29 D02 D05 D08 D11 D12 D14 D16 D17 D20 D23 D24 D25 D27 D28 D29
C15	R3·R4·R5·R7·R8·R9·R12·R15·R16·R18·R20·R21·R24·R27·R30·D1·D4·D7·D10·D11·D13·D15·D16·D19·D22·D23·D24·D26·D27·D28
C16	R0·R4·R5·R8·R12·R13·R17·R19·R21·R22·R24·R26·R29·R30·D1·D2·D5·D7·D9·D10·D12·D14·D18·D19·D23·D26·D27·D31
C17	R1·R5·R6·R9·R13·R14·R18·R20·R22·R25·R27·R30·R31·D0·D1·D4·D6·D8·D9·D11·D13·D17·D18·D22·D25·D26·D30
C18	R2·R6·R7·R10·R14·R15·R19·R21·R23·R24·R26·R28·R31·D0·D3·D5·D7·D8·D10·D12·D16·D17·D21·D24·D25·D29
C19	R3·R7·R8·R11·R15·R16·R20·R22·R24·R25·R27·R29·D2·D4·D6·D7·D9·D11·D15·D16·D20·D23·D24·D28
C20	R4·R8·R9·R12·R16·R17·R21·R23·R25·R26·R28·R30·D1·D3·D5·D6·D8·D10·D14·D15·D19·D22·D23·D27
C21	R5·R9·R10·R13·R17·R18·R22·R24·R26·R27·R29·R31·D0·D2·D4·D5·D7·D9·D13·D14·D18·D21·D22·D26
C22	R0·R9·R11·R12·R14·R16·R18·R19·R23·R24·R26·R27·R29·R31·D0·D2·D4·D5·D7·D8·D12·D13·D15·D17·D19·D20·D22·D31

9

C23	R0·R1·R6·R9·R13·R15·R16·R17·R19·R20·R26·R27·R29·R31·D0·D2· D4·D5·D11·D12·D14·D15·D16·D18·D22·D25·D30·D31
C24	R1·R2·R7·R10·R14·R16·R17·R18·R20·R21·R27·R28·R30·D1·D3·D4· D10·D11·D13·D14·D15·D17·D21·D24·D29·D30
C25	R2·R3·R8·R11·R15·R17·R18·R19·R21·R22·R28·R29·R31·D0·D2·D3· D9·D10·D12·D13·D14·D16·D20·D23·D28·D29
C26	R0·R3·R4·R6·R10·R18·R19·R20·R22·R23·R24·R25·R26·R28·R31· D0·D3·D5·D6·D7·D8·D9·D11·D12·D13·D21·D25·D27·D28·D31
C27	R1·R4·R5·R7·R11·R19·R20·R21·R23·R24·R25·R26·R27·R29·D2·D4· D5·D6·D7·D8·D10·D11·D12·D20·D24·D26·D27·D30
C28	R2·R5·R6·R8·R12·R20·R21·R22·R24·R25·R26·R27·R28·R30·D1·D3· D4·D5·D6·D7·D9·D10·D11·D19·D23·D25·D26·D29
C29	R3·R6·R7·R9·R13·R21·R22·R23·R25·R26·R27·R28·R29·R31·D0·D2· D3·D4·D5·D6·D8·D9·D10·D18·D22·D24·D25·D28
C30	R4·R7·R8·R10·R14·R22·R23·R24·R26·R27·R28·R29·R30·D1·D2·D3· D4·D5·D7·D8·D9·D17·D21·D23·D24·D27
C31	R5·R8·R9·R11·R15·R23·R24·R25·R27·R28·R29·R30·R31·D0·D1·D2· D3·D4·D6·D7·D8·D16·D20·D22·D23·D26

The above shows operational expressions in which the input data is made up of 32 bits (4 bytes: D0-D31). If input data length is different from 32 bits, other operational expressions different from the above ones have to be used. For example, in a case where the input data is made up of 8 bits (1 byte: D0-D07), an output data from each flip-flop at a time point when D7 has been input (that is, eight shifts) becomes each of the required operational expressions. Also, in a case where the input data is made up of 64 bits (8 byte: D0-D63), an output data from each flip-flop at a time point when D63 has been input (that is, 64 shifts) becomes each of the required operational expressions.

The above is all of general explanation about the CRC method and operational expression.
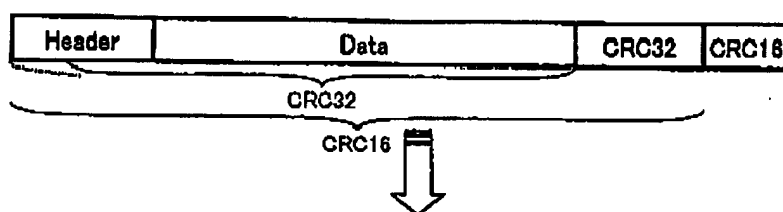
Next, a supplementary explanation relating to the present invention will be given.

10

【Data Format】
To begin with, a general data format will be explained.
The communications data is generally made up of header information, data, and a CRC arranged before CRC32 in the data format, the CRC16 arithmetic, as described below. The CRC uses generally either of 32 bits and 16 bits in accordance with a requirement for a system (Error detecting methods other than CRC will be omitted from this explanation, since they are not relevant to configurations of the present invention).
The present invention is preferably used in a system (such as an InfiniBand or a like) in which two or more CRC arithmetic operation results are required.

Header	Data	CRC32	CRC16

CRC32

CRC16

Example: 4-byte (32-bit) transmission

1byte

#1	data0	data1	data2	data3
#2	data4	data5	data6	data7
#3	data8	data9	data10	data11

#n−2	data n−5	data n−4	data n−3	data n−2
#n−1	data n−1	data n	CRC32	CRC32
#n	CRC32	CRC32	CRC16	CRC16

The CRC method is one of error detection methods for detecting error on the data to be transmitted. That is, in this CRC method, when the above-mentioned data are transmitted, it is usable for error detection on data 0 to data n.
In processing of a CRC16 arithmetic operation, a CRC32 arithmetic operation result is treated as data, and as a result, error detection on data including a CRC32 arithmetic operation result is performed.
For this reason, the CRC16 arithmetic operation needs the CRC32 arithmetic operation result, whereas a CRC32 arithmetic operation does not need a CRC16 arithmetic operation result.
To put it simply, in a data transmitting/receiving, the difference is only whether the CRC32 arithmetic operation result is transmitted earlier or later than the CRC16 arithmetic operation result. If the CRC16 arithmetic operation result is

11

arranged before the CRC32 arithmetic operation result in the data format, the CRC16 arithmetic operation result is to be used in CRC32 arithmetic operation.

Note: a reason for adding two kinds of CRC code to data will be explained with reference to an example of InfiniBand.

others   InfiniBand
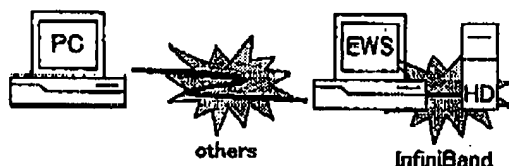
With a system configuration as shown in the above figure (a conventional protocol such as TCP-IP is used as a communications protocol for data communications carried out between a PC (Personal Computer) and a server (EWS), whereas an InfiniBand protocol is used as a communications protocol for data communications carried out between the server (EWS) and an HD (Hard Disk)),

When data access (reading) is made from the PC to HD through a server (EWS), data read from the HD and configured in the foregoing data format is transmitted to the EWS in accordance with the InfiniBand protocol. Next, when the EWS has received data from the HD, the EWS transmits the received data to the PC, without performing a further CRC arithmetic operation. That is, data obtained by removing a header prepared specifically for the InfiniBand protocol and the CRC16 arithmetic operation result from the received data is transmitted from the EWS to the PC. Then, if the CRC32 arithmetic operation result is not added to the received data, the CRC32 arithmetic operation has to be preformed. This means savings in time and power.

[Background of the Invention]

As described above, it is necessary to add the CRC arithmetic operation result to an end of the data to be transmitted.

Generally, in data communications, in order to transmit data accurately to a receiver, continuous transmission from a beginning to an end of the data transmission (in the case of a packet communication, during transmission of one packet) is required. Therefore, a time lag (interruption) between the end of the data and the CRC arithmetic operation result should be avoided. Moreover, to obtain the CRC arithmetic operation result, time being equivalent to at least one clock is necessary.

12

For this reason, the CRC arithmetic operation result has to follow continuously the end of the data to be transmitted, by inserting latches between Data paths. Furthermore, in a case of having two kinds of CRC arithmetic operation results, at least two latches have to be inserted between Data paths, since time being equivalent to at least two clocks is necessary (in order to use one CRC arithmetic operation result in another arithmetic operation).

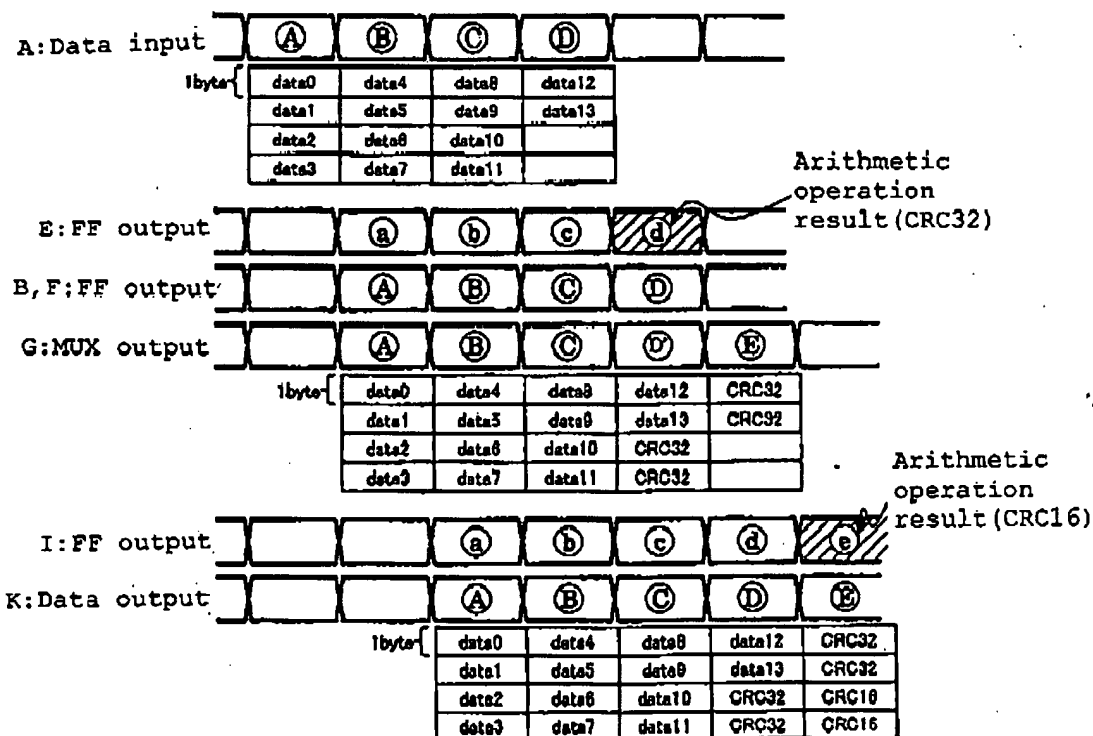High-speed signal processing also in data communications has become indispensable, as high-speed operations of CPUs (Central Processing Units) have been achieved in recent years. To achieve high-speed signal processing, it may be preferable to increase a data transmission speed and/or a width of a bus. In addition to these, an increase in the processing speed within a signal processing circuit is essential.

13

[Conventional Circuit Configuration]
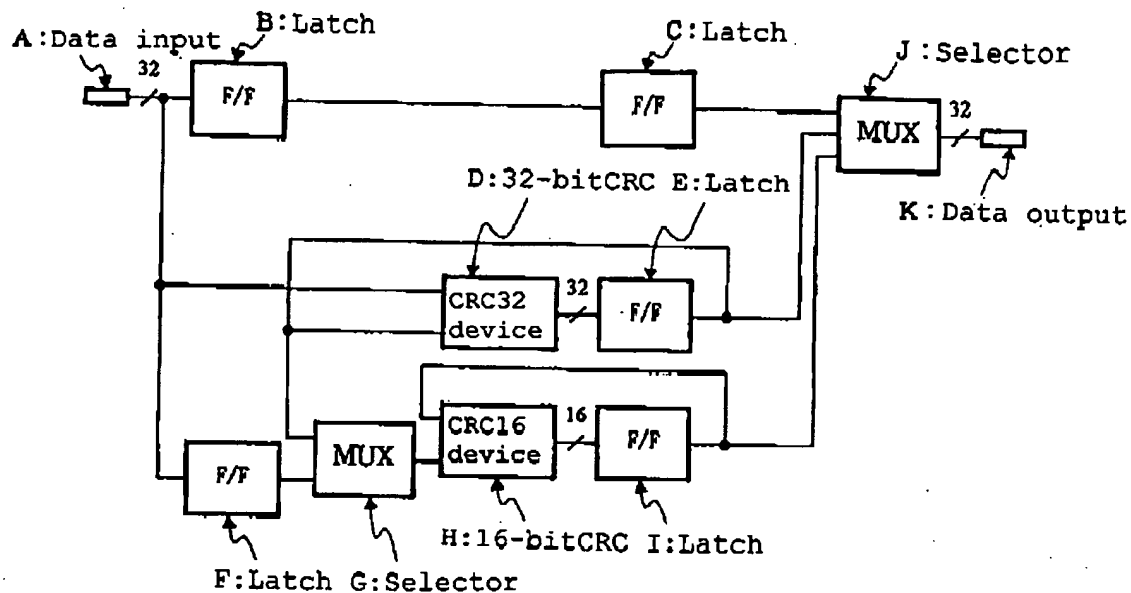
A: Data inputting section
B, C: Latch (32-bit flip-flop)
       for adjusting operation timing in data path
D: CRC32 arithmetic operation device
E: Latch (32-bit flip-flop)
       for latching CRC32 arithmetic operation result
F: Latch (16-bit flip-flop) for adjusting operation
       timing of CRC16 arithmetic operation device
G: Selector circuit for selecting either of latched
       input data and CRC32 arithmetic operation result
H: CRC16 arithmetic operation device
I: Latch (16-bit flip-flop)
       for latching CRC16 arithmetic operation result
J: Output selector
K: Data outputting section

14

Thirty-two bits (4 bytes) of data output from a data
inputting section A are input to a latch B, a CRC32 arithmetic
operation device D, and a latch F which is used to adjust operation
timing of a CRC16 arithmetic operation device. A CRC32 arithmetic
operation result (an arithmetic operation result of 32 bits
output from the CRC32 arithmetic operation device D) is input
to an output selector J, and a selector circuit G through a latch
E. In the selector circuit G, either of the input data and the
CRC32 arithmetic operation result is selected as data to be input
to a CRC16 arithmetic operation device H. A CRC16 arithmetic
operation result (an arithmetic operation result of 16 bits
output from the CRC16 arithmetic operation device H) is input
to the output selector J through a latch I.

A:Data input

Ⓐ	Ⓑ	Ⓒ	Ⓓ		

1byte{

data0	data4	data8	data12
data1	data5	data9	data13
data2	data6	data10	
data3	data7	data11	

Arithmetic operation result (CRC32)

E:FF output

	ⓐ	ⓑ	ⓒ	ⓓ	

B,F:FF output

	Ⓐ	Ⓑ	Ⓒ	Ⓓ	

G:MUX output

	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ

1byte{

data0	data4	data8	data12	CRC32
data1	data5	data9	data13	CRC32
data2	data6	data10	CRC32	
data3	data7	data11	CRC32	

Arithmetic operation result (CRC16)

I:FF output

		ⓐ	ⓑ	ⓒ	ⓓ	ⓔ

K:Data output

		Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ

1byte{

data0	data4	data8	data12	CRC32
data1	data5	data9	data13	CRC32
data2	data6	data10	CRC32	CRC16
data3	data7	data11	CRC32	CRC16

Let it be assumed that data as shown in the above timing
chart is input from the data inputting section A, and that an
end part Ⓓ of the input data is made up of only 2 bytes (16
bits). The first CRC32 arithmetic operation is performed by using
a first part Ⓐ of the input data and the initial value of the
latch E. The latch E latches the first arithmetic operation result
obtained from the CRC32 arithmetic operation device D. After
this, the second CRC32 arithmetic operation is performed by using
a second part Ⓑ of the input data and the first CRC32 arithmetic

15

operation result(data latched in the latch E). By repeating the above CRC32 arithmetic operations, the CRC32 code bit ⓓ can finally be obtained.
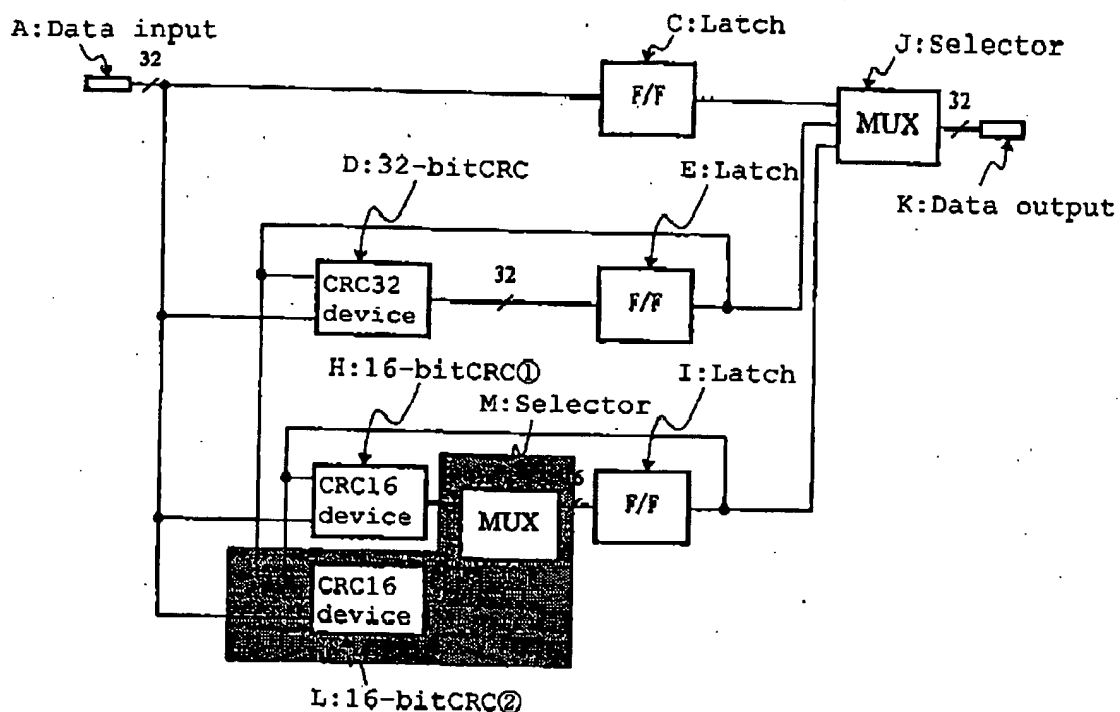
Next, it is necessary to add continuously the CRC32 arithmetic operation result ⓓ to the end part ⓓ of the input data fed from the data inputting section A and then to feed the added data to the CRC16 arithmetic operation device H.

Then, in a case where the end part ⓓ of the input data is made up of only 2 bytes, it is necessary to input the CRC32 arithmetic operation result separately at two timings, as shown in the above timing chart. For this reason, it is necessary to delay the input data by one clock. Therefore, the latch F is provided on an input side of selector circuit (MUX section) G.

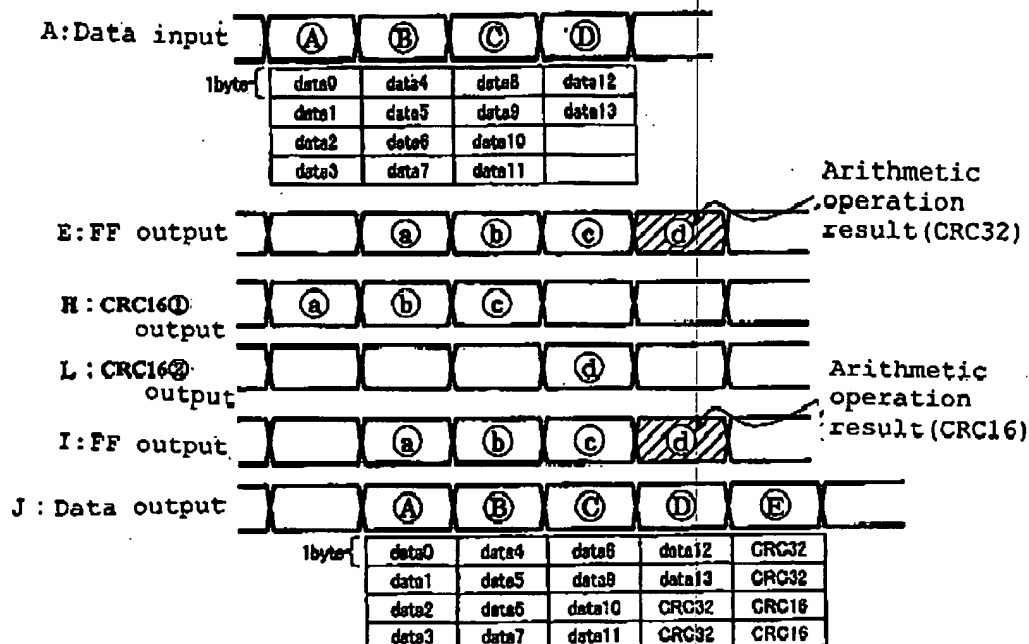Thus, the CRC16 code bit ⓔ can finally be obtained.

With the conventional circuit configuration described above, a time delay being equivalent to two clocks occurs between inputting of data and outputting of data.

16

[Circuit Configuration of the Invention]

A:Data input
32
C:Latch
J:Selector
F/F
MUX
32
K:Data output
D:32-bitCRC
E:Latch
CRC32 device
32
F/F
H:16-bitCRC①
I:Latch
M:Selector
CRC16 device
MUX
F/F
CRC16 device
L:16-bitCRC②

A: Data inputting section
C: Latch (32-bit flip-flop) for adjusting operation timing in data path
D: CRC32 arithmetic operation device
E: Latch (32-bit flip-flop) for latching CRC32 arithmetic operation result
H: CRC16 arithmetic operation device ①
I: Latch (16-bit flip-flop) for latching CRC16 arithmetic operation result
J: Output selector
K: Data outputting section
L: CRC16 arithmetic operation device ② (for reducing latency)
M: Output selector

Thirty-two bits (4 bytes) of data output from a data inputting section A are input to a latch C, a CRC32 arithmetic operation device D, a CRC16 arithmetic operation device ① H, and a CRC16 arithmetic operation device ② L. Arithmetic operation results obtained from these arithmetic operation devices are input to the output selector J through respective latches E and I.

17

| A:Data input | Ⓐ | Ⓑ | Ⓒ | Ⓓ | | |

1byte	data0	data4	data8	data12
	data1	data5	data9	data13
	data2	data6	data10	
	data3	data7	data11	

Arithmetic operation result(CRC32)

| E:FF output | | ⓐ | ⓑ | ⓒ | ⓓ | |

| H : CRC16① output | ⓐ | ⓑ | ⓒ | | | |

| L : CRC16② output | | | | ⓓ | | |

Arithmetic operation result(CRC16)

| I:FF output | | ⓐ | ⓑ | ⓒ | ⓓ | |

| J : Data output | | Ⓐ | Ⓑ | Ⓒ | Ⓓ | Ⓔ |

1byte	data0	data4	data8	data12	CRC32
	data1	data5	data9	data13	CRC32
	data2	data6	data10	CRC32	CRC16
	data3	data7	data11	CRC32	CRC16

Let it be assumed that data as shown in the above timing chart is input from the data inputting section A (same as the conventional example).

The first CRC32 arithmetic operation is performed by using the first part (A) of the input data and the initial value of the latch E. The latch E latches the first arithmetic operation result obtained from the CRC32 arithmetic operation device D. After this, the second CRC32 arithmetic operation is performed by using the second part (B) of the input data and the first CRC32 arithmetic operation result(data latched in the latch E). By repeating the above CRC32 arithmetic operations, the CRC32 code bit (d) can finally be obtained.

Next, the CRC16 arithmetic operation device ① H performs the first CRC16 arithmetic operation by using the first part (A) of the input data and the initial value of the latch I. The latch I latches the first arithmetic operation result obtained from a CRC16 arithmetic operation device ① H. After this, the second CRC32 arithmetic operation is performed by using the second part (B) of the input data and the first CRC16 arithmetic operation result(data latched in the latch I).

18

The above CRC16 arithmetic operations are repeated up to the data part (C), immediately (one clock) before the end part (D) of the input data, the CRC32 code bit (d) can finally be obtained.

When the end part (D) of the input data is detected, an output selector (MUX) M selects the CRC16 arithmetic operation device ② L, and the CRC16 code bit (d) can finally be obtained from the CRC16 arithmetic operation device ② L.

The CRC16 arithmetic operation device ② L inputs the end part (D) of the input data, the CRC16 ① arithmetic operation result (c) obtained from the CRC16 arithmetic operation device ① H through the latch I, and the CRC32 arithmetic operation result (c) obtained from the CRC32 arithmetic operation device D through the latch E, in order to expedite timing.

At this stage, the CRC32 arithmetic operation result is obtained by performing the CRC32 arithmetic operation using the end part (D) of the input data and the immediately preceding CRC32 arithmetic operation result.

Accordingly, with the above configuration of the present invention having feature in that the CRC32 arithmetic operation is included (incorporated) in the CRC16 arithmetic operation, it is possible to perform the CRC16 arithmetic operation, without using (waiting for) the CRC32 arithmetic operation result.

The CRC16 arithmetic operation result can be obtained one clock (MIN.) earlier, compared to that in the conventional configuration, since it is not necessary to wait for the CRC32 arithmetic operation result. Only one clock delay occurs even on the side of the data path.

With the circuit configuration of the present invention, a time delay being equivalent to one clock occurs between inputting of data and outputting of data. This means reduction of latency by one clock (MIN.), compared to the conventional circuit configuration.

19

[Operational Expression Producing Method]
The CRC arithmetic operation devices, which are used in a conventional example and an embodiment according to the present invention, use the following operational expression.

CRC32 arithmetic operation device D:
The device D uses the operational expression described on pages 6 to 8 of this Document.

CRC16 arithmetic operation device H:
The device H utilizes the generating circuit described on page 5 and uses a set of output data being output from each of the flip-flops when 32 bits of data were shifted. In fact, the operational expression described on page 5 is an example in a case where 16 bits of data are shifted.

CRC16 arithmetic operation device L:
The device L produces newly an operational expression by using the following method:

As described under the section "circuit configuration of the Invention", if the CRC16 arithmetic operation is performed after the CRC32 arithmetic operation result was obtained, a time delay being equivalent to one clock occurs inevitably.
To solve this problem, it is preferable that the CRC32 arithmetic operation is simultaneously performed, when the CRC16 arithmetic operation is performed, whereby it becomes possible to acquire simultaneously the CRC32 and the CRC16 arithmetic operation results, without occurrence of a time delay.

Therefore, in order to avoid such a time delay, a new operational expression is produced and used according to the procedures as below:
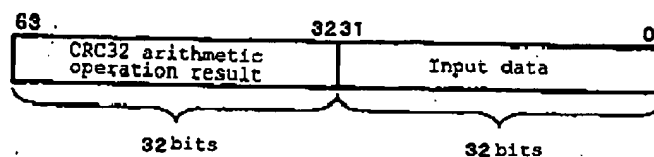
① data being of 64 bits in length.
In the CRC16 arithmetic operation device, the operational expression is produced using input data (32 bits) and an immediately preceding arithmetic operation result (16 bits).
With the conventional method, CRC16 code bit is acquired by adding CRC32 code bit (32 bits) to an end part of the input data.
The CRC16 arithmetic operation device ② L incorporated in the present invention inputs simultaneously the input data and CRC32 arithmetic operation result (obtained immediately before a final CRC32 arithmetic operation result). That is, the operational expression is produced as 64 bits of the input data. At this stage, original input data (D) as lower-order bits and

20

the CRC arithmetic operation result as higher-order bits are respectively input.

② Operational Expression Production-I

First, a CRC16 operational expression on input data being of 64 bits in length is produced.

This is the output data from each of the flip-flops making up the CRC16 generating circuit described on page 5, when 64 bits of data is shifted.

C0	R2·R4·R5·R8·R9·R11·R12·R13·R14·D1·D2·D3·D4·D6·D7·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·D32·D34·D35·D37·D40·D42·D43·D48·D50·D51·D55·D59·D63
C1	R2·R3·R4·R6·R8·R10·R11·R15·D0·D4·D5·D7·D9·D11·D12·D13·D16·D20·D21·D22·D25·D26·D28·D29·D30·D32·D33·D35·D36·D37·D39·D40·D41·D43·D47·D48·D49·D51·D54·D55·D58·D59·D62·D63
C2	R0·R3·R4·R5·R7·R9·R11·R12·D3·D4·D6·D8·D10·D11·D12·D15·D19·D20·D21·D24·D25·D27·D28·D29·D31·D32·D34·D35·D36·D38·D39·D40·D42·D46·D47·D48·D50·D53·D54·D57·D58·D61·D62
C3	R1·R2·R6·R9·R10·R11·R14·D1·D4·D5·D6·D9·D13·D14·D17·D22·D23·D24·D27·D28·D29·D30·D32·D33·D38·D39·D40·D41·D42·D43·D45·D46·D47·D48·D49·D50·D51·D52·D53·D55·D56·D57·D59·D60·D61·D63

C4	R2·R3·R7·R10·R11·R12·R15·D0·D3·D4·D5·D8·D12·D13·D16·D21·D22·D23·D26·D27·D28·D29·D31·D32·D37·D38·D39·D40·D41·D42·D44·D45·D46·D47·D48·D49·D50·D51·D52·D54·D55·D56·D58·D59·D60·D62
C5	R0·R3·R4·R8·R11·R12·R13·D02·D03·D4·D7·D11·D12·D15·D20·D21·D22·D25·D26·D27·D28·D30·D31·D36·D37·D38·D39·D40·D41·D43·D44·D45·D46·D47·D48·D49·D50·D51·D53·D54·D55·D57·D58·D59·D61
C6	R1·R4·R5·R9·R12·R13·R14·D1·D2·D3·D6·D10·D11·D14·D19·D20·D21·D24·D25·D26·D27·D29·D30·D35·D36·D37·D38·D39·D40·D42·D43·D44·D45·D46·D47·D48·D49·D50·D52·D53·D54·D56·D57·D58·D60
C7	R2·R5·R6·R10·R13·R14·R15·D0·D1·D2·D5·D09·D10·D13·D18·D19·D20·D23·D24·D25·D26·D28·D29·D34·D35·D36·D37·D38·D39·D41·D42·D43·D44·D45·D46·D47·D48·D49·D51·D52·D53·D55·D56·D57·D59
C8	R3·R6·R7·R11·R14·R15·D0·D1·D4·08·D9·D12·D17·D18·D19·D22·D23·D24·D25·D27·D28·D33·D34·D35·D36·D37·D38·D40·D41·D42·D43·D44·D45·D46·D47·D48·D50·D51·D52·D54·D55·D56·D58
C9	R4·R7·R8·R12·R15·D0·D3·D7·D8·D11·D16·D17·D18·D21·D22·D23·D24·D26·D27·D32·D33·D34·D35·D36·D37·D39·D40·D41·D42·D43·D44·D45·D46·D47·D49·D50·D51·D53·D54·D55·D57
C10	R0·R5·R08·R9·R13·D2·D6·D7·D10·D15·D16·D17·D20·D21·D22·D23·D25·D26·D31·D32·D33·D34·D35·D36·D38·D39·D40·D41·D42·D43·D44·D45·D46·D48·D49·D50·D52·D53·D54·D56
C11	R0·R1·R6·R9·R10·R14·D1·D5·D6·D9·D14·D15·D16·D19·D20·D21·D22·D24·D25·D30·D31·D32·D33·D34·D35·D37·D38·D39·D40·D41·D42·D43·D44·D45·D47·D48·D49·D51·D52·D53·D55
C12	R0·R1·R4·R5·R7·R8·R9·R10·R12·R13·R14·R15·D0·D1·D2·D3·D5·D6·D7·D8·D10·D11·D14·D15·D17·D21·D22·D23·D24·D26·D30·D33·D35·D36·D38·D39·D41·D44·D46·D47·D52·D54·D55·D59·D63
C13	R1·R2·R5·R6·R8·R9·R10·R11·R13·R14·R15·D0·D1·D2·D4·D5·D6·D7·D9·D10·D13·D14·D16·D20·D21·D22·D23·D25·D29·D32·D34·D35·D37·D38·D40·D43·D45·D46·D51·D53·D54·D58·D62
C14	R0·R2·R3·R6·R7·R9·R10·R11·R12·R14·R15·D0·D1·D3·D4·D5·D6·D8·D9·D12·D13·D15·D19·D20·D21·D22·D24·D28·D31·D33·D34·D36·D37·D39·D42·D44·D45·D50·D52·D53·D57·D61
C15	R1·R3·R4·R7·R8·R10·R11·R12·R13·R15·D0·D2·D3·D4·D5·D7·D8·D11·D12·D14·D18·D19·D20·D21·D23·D27·D30·D32·D33·D35·D36·D38·D41·D43·D44·D49·D51·D52·D56·D60

③. Replacement of data
Operational expressions described on pages 8-9 are substituted into operational expressions obtained in "②", since D63-D31 are the CRC32 arithmetic operation results, as clear from "①".

Substituting

$C0 =$ R2·R4·R5·R8·R9·R11·R12·R13·R14·
D1·D2·D3·D4·D6·D7·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·
R5·R8·R9·R11·R15·R23·R24·R25·R27·R28·R29·R30·R31·
D0·D1·D2·D3·D4·D6·D7·D8·D16·D20·D22·D23·D26
R1·R4·R5·R7·R11·R19·R20·R21·R23·R24·R25·R26·R27·R29·
D2·D4·D5·D6·D7·D8·D10·D11·D12·D20·D24·D26·D27·D30
R0·R1·R6·R9·R13·R15·R16·R17·R19·R20·R26·R27·R29·R31·
D0·D2·D4·D5·D11·D12·D14·D15·D16·D18·D22·D25·D30·D31
R3·R7·R8·R11·R15·R16·R20·R22·R24·R25·R27·R29·
D2·D4·D6·D7·D9·D11·D15·D16·D20·D23·D24·D28
R2·R6·R7·R10·R14·R15·R19·R21·R23·R24·R26·R28·R31·
D0·D3·D5·D7·D8·D10·D12·D16·D17·D21·D24·D25·D29
R0·R4·R5·R8·R12·R13·R17·R19·R21·R22·R24·R26·R29·R30·
D1·D2·D5·D7·D9·D10·D12·D14·D18·D19·D23·D26·D27·D31
R0·R1·R3·R4·R9·R12·R14·R15·R16·R17·R20·R24·R25·R26·R27·R28·R31·
D0·D3·D4·D5·D6·D7·D11·D14·D15·D16·D17·D19·D22·D27·D28·D30·D31
R0·R2·R3·R5·R9·R13·R14·R16·R19·R26·R28·R29·R31·
D0·D2·D3·D5·D12·D15·D17·D18·D22·D26·D28·D29·D31
R0·R1·R3·R4·R8·R10·R11·R17·R22·R28·R31·
D0·D3·D8·D9·D14·D19·D20·D21·D23·D27·D28·D30·D31
R0·R1·R3·R4·R5·R6·R7·R10·R13·R19·R20·R21·R24·R28·R29·
D2·D3·D7·D10·D11·D12·D18·D21·D24·D25·D26·D27·D28·D30·D31
R1·R2·R3·R7·R8·R9·R10·R14·R15·R17·R18·R19·R25·R27·R31·
D0·D4·D6·D12·D13·D14·D16·D17·D21·D22·D23·D24·D28·D29·D30
R0·R2·R6·R7·R8·R9·R13·R14·R16·R17·R18·R24·R26·R30·R31·
D0·D1·D5·D7·D13·D14·D15·D17·D18·D22·D23·D24·D25·D29·D30·D31
R0·R6·R9·R10·R12·R16·R24·R25·R26·R28·R29·R30·R31·
D0·D1·D2·D3·D5·D6·D7·D15·D19·D21·D22·D25·D31

Deleting same terms

$C0 =$ Z2·Z4·Z5·Z8·Z9·Z11·Z12·Z13·Z14·
R1·R4·R5·R6·R9·R10·R13·R14·R19·R20·R22·R24·R28·R31·
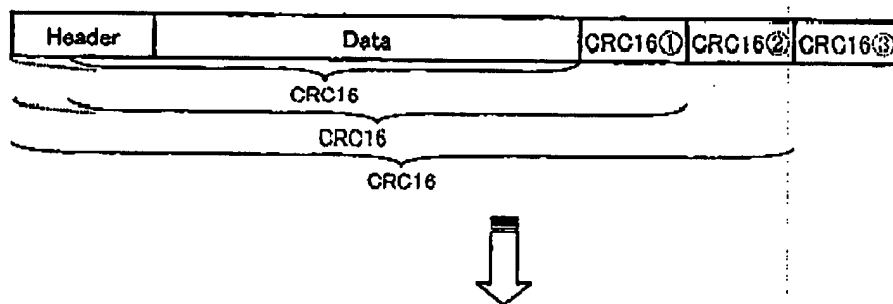D0·D1·D2·D4·D6·D9·D10·D12·D13·D19·D20·D21·D25·D27·D29·D30·D31·

It is possible to obtain simultaneously final CRC32 and CRC16 arithmetic operation results, by performing the above processing on all the operational expressions.

1

Three CRC16 arithmetic operation devices are incorporated in a circuit of a second embodiment of the present invention. A data format, a circuit configuration, operation timing and operational expressions in the second embodiment will be described below.

【Data Format】

To begin with, a data format in a case where three CRC16 arithmetic operation devices are required will be explained.

Please note that, even if another arithmetic operation (CRC32 arithmetic operation device) being different from the CRC16 arithmetic operation is used, new operational expressions can be produced using a same circuit configuration in accordance with an algorithm described below.

Header	Data			CRC16①	CRC16②	CRC16③

CRC16

CRC16

CRC16

Example: 4-byte (32-bit) transmission

1byte

#1	data0	data1	data2	data3
#2	data4	data5	data6	data7
#3	data8	data9	data10	data11

⋮

#n-2	data n-4	data n-3	data n-2	data n-1
#n-1	data n	CRC16①	CRC16①	CRC16②
#n	CRC16②	CRC16③	CRC16③	

In processing of a CRC16 ② arithmetic operation, a CRC16 ① arithmetic operation result is treated as data, and as a result, error detection on data including a CRC16① arithmetic operation result is performed. In addition, in processing of a CRC16 ③ arithmetic operation, the CRC16 ① and CRC16 ② arithmetic operation results are treated as data, and as a result, error detection on data including the CRC16 ② arithmetic operation result is performed.

2

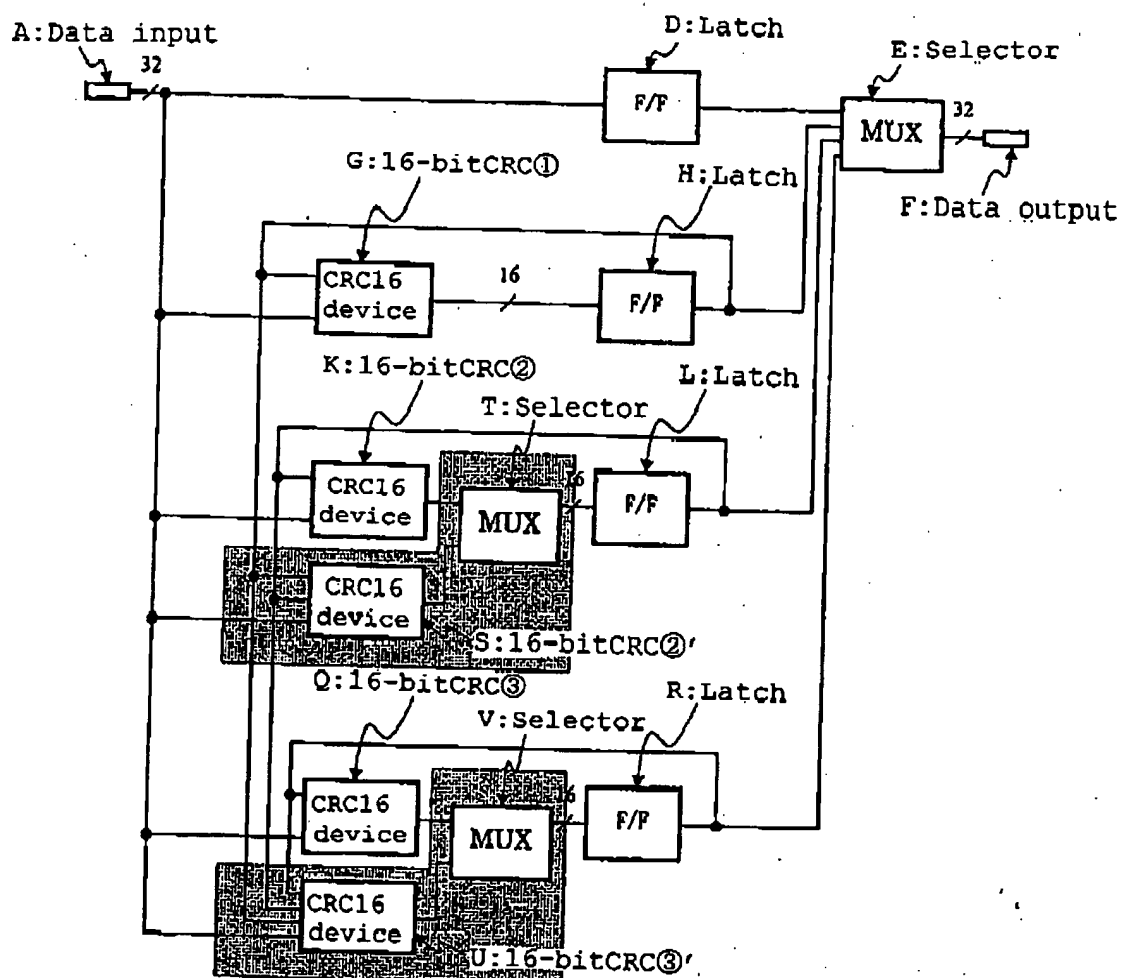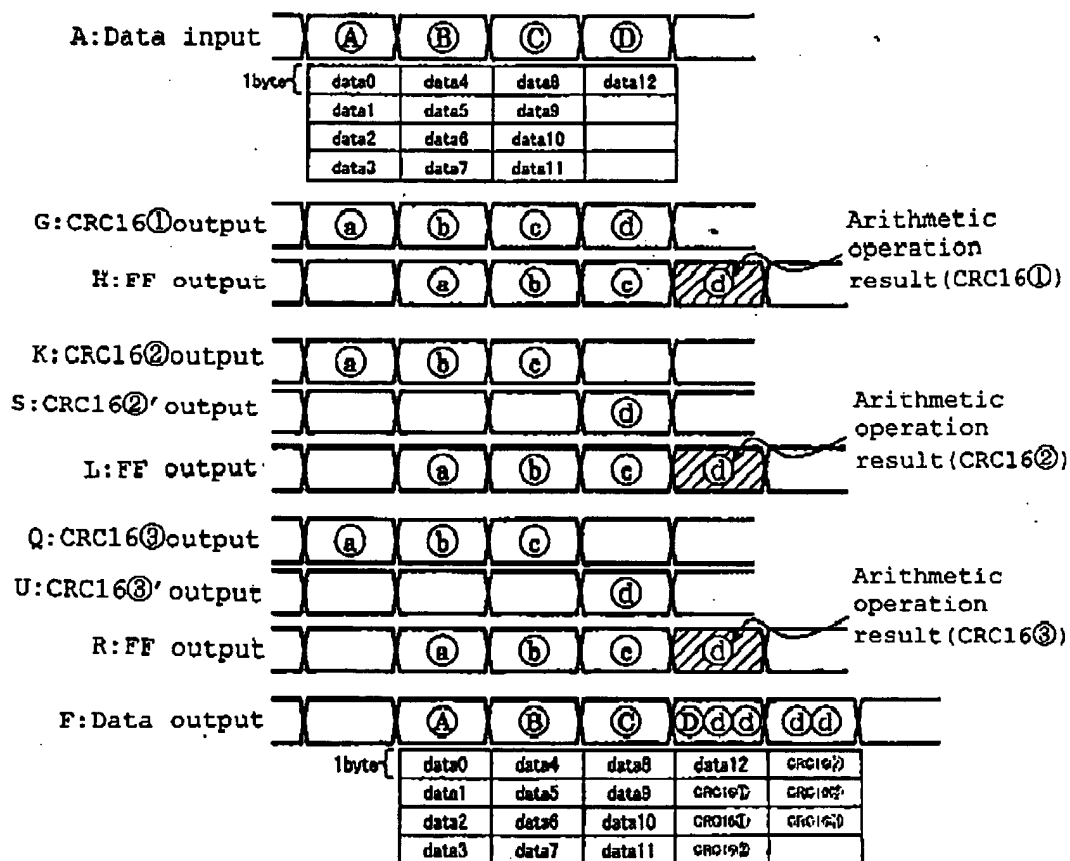[Circuit Configuration according to the Conventional Method]
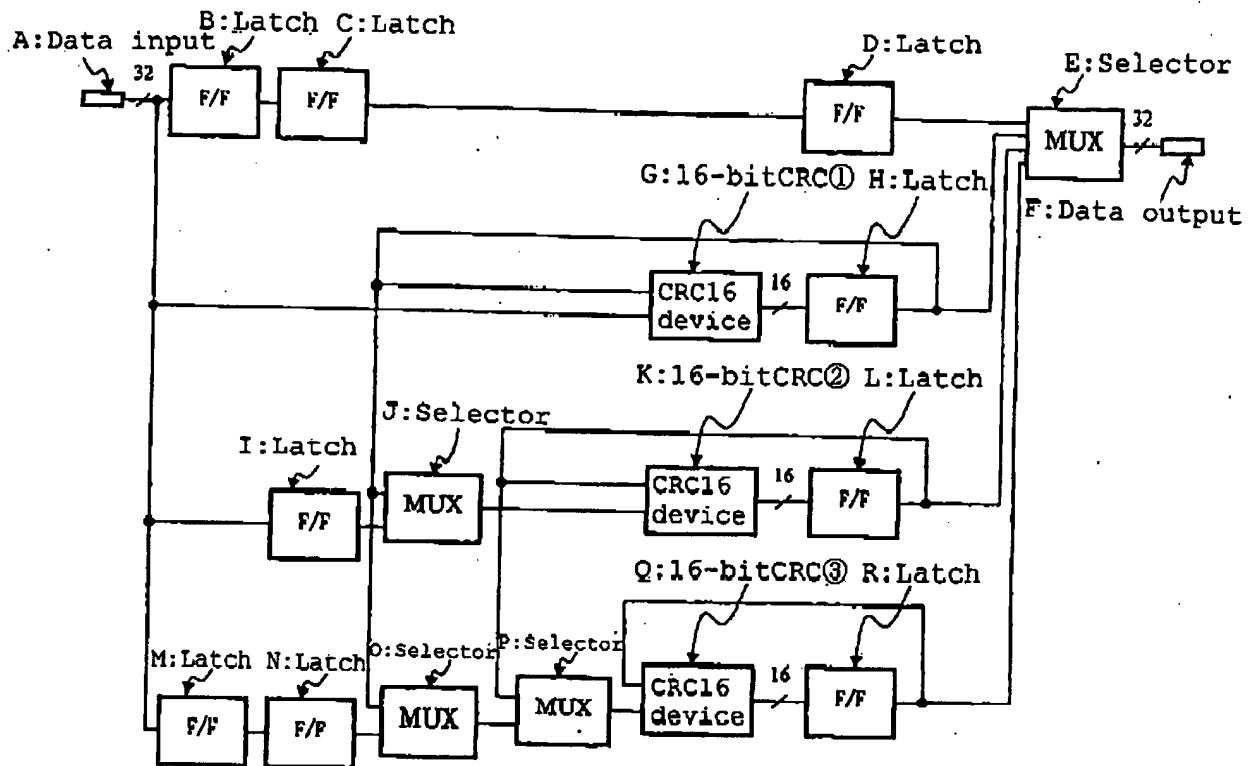
A: Data inputting section
B, C, D: Latch (32-bit flip-flop)
             for adjusting operation timing in data path
E: Output selector
F: Data outputting section
G: CRC16 arithmetic operation device ①
H: Latch (16-bit flip-flop)
             for latching CRC16 ① arithmetic operation result
I: Latch (32-bit flip-flop) for adjusting operation
             timing of CRC16 ② arithmetic operation device
J, O: Selector circuit for selecting either of latched
             input data and CRC16 ① arithmetic operation result
K: CRC16 arithmetic operation device ②
             (for reducing latency)
L: Latch (16-bit flip-flop)
             for latching CRC16 ② arithmetic operation result
M, N: Latch (32-bit flip-flop) for adjusting operation
             timing of CRC16 ② arithmetic operation device
P: Selector circuit for selecting either of output from
             selector circuit O and CRC16 ② arithmetic operation
             result
Q: CRC16 arithmetic operation device ③
R: Latch (16-bit flip-flop)
             for latching CRC16 ③ arithmetic operation result

3

A:Data input

data0	data4	data8	data12
data1	data5	data9	
data2	data6	data10	
data3	data7	data11	

1byte

G:CRC16①output

H:FF output — Arithmetic operation result(CRC16①)

B,I,M:FF output

J:Selector output

1byte

data0	data4	data8	data12
data1	data5	data9	CRC16①
data2	data6	data10	CRC16①
data3	data7	data11	

K:CRC16②output — Arithmetic operation result(CRC16②)

L:FF output

C,N:FF output

P:Selector output

1byte

data0	data4	data8	data12	CRC16②
data1	data5	data9	CRC16①	
data2	data6	data10	CRC16①	
data3	data7	data11	CRC16②	

Arithmetic operation result (CRC16③)

Q:CRC16③output

R:FF output

F:Data output

1byte

data0	data4	data8	data12	CRC16②
data1	data5	data9	CRC16①	CRC16①
data2	data6	data10	CRC16①	CRC16③
data3	data7	data11	CRC16②	

4

[Circuit Configuration according to Second Embodiment]

A:　Data inputting section
D:　Latch (32-bit flip-flop)
　　for adjusting operation timing in data path
E:　Output selector
F:　Data outputting section
G:　CRC16 arithmetic operation device ①
H:　Latch (16-bit flip-flop)
　　for latching CRC16 ① arithmetic operation result
K:　CRC16 arithmetic operation device ②
L:　Latch (16-bit flip-flop)
　　for latching CRC16 ② arithmetic operation result
Q:　CRC16 arithmetic operation device ③
R:　Latch (16-bit flip-flop)
　　for latching CRC16 ③ arithmetic operation result
S:　CRC16 arithmetic operation device ②'
　　(for reducing latency)
T:　Output selector for selecting either of CRC16
　　arithmetic operation devices ②, ②'
U:　CRC16 arithmetic operation device ③'
　　(for reducing latency)
V:　Output selector for selecting either of CRC16
　　arithmetic operation devices ③, ③'

5

A:Data input	Ⓐ	Ⓑ	Ⓒ	Ⓓ		

1byte:

data0	data4	data8	data12
data1	data5	data9	
data2	data6	data10	
data3	data7	data11	

G:CRC16①output	ⓐ	ⓑ	ⓒ	ⓓ		
H:FF output		ⓐ	ⓑ	ⓒ	ⓓ	

Arithmetic operation result(CRC16①)

K:CRC16②output	ⓐ	ⓑ	ⓒ			
S:CRC16②′ output				ⓓ		
L:FF output		ⓐ	ⓑ	ⓒ	ⓓ	

Arithmetic operation result(CRC16②)

Q:CRC16③output	ⓐ	ⓑ	ⓒ			
U:CRC16③′ output				ⓓ		
R:FF output		ⓐ	ⓑ	ⓒ	ⓓ	

Arithmetic operation result(CRC16③)

F:Data output		Ⓐ	Ⓑ	Ⓒ	ⓓⓓ	ⓓⓓ

1byte:

data0	data4	data8	data12	CRC16⑦
data1	data5	data9	CRC16①	CRC16②
data2	data6	data10	CRC16①	CRC16③
data3	data7	data11	CRC16②	

Let it be assumed that data as shown in the above timing chart is input from the data inputting section A (same as the conventional example).

The first CRC16 ① arithmetic operation is performed by using the first part Ⓐ of the input data and the initial value of the latch H. The latch H latches the first arithmetic operation result obtained from the CRC16 ① arithmetic operation device G. After this, the second CRC16 ① arithmetic operation is performed by using the second part Ⓑ of the input data and the first CRC16 ① arithmetic operation result(data latched in the latch H). By repeating the above CRC16 ① arithmetic operations, the CRC16 ① code bit ⓓ can finally be obtained.

Next, in a same manner as described above, the CRC16 arithmetic operation device ② K also performs the first CRC16 ② arithmetic operation by using the first part Ⓐ of the input data and the initial value of the latch L.

The latch L latches the first arithmetic operation result obtained from a CRC16 arithmetic operation device ② K. The above

6

CRC16 ② arithmetic operations are repeated up to the data part Ⓒ immediately (one clock) before the end part Ⓓ of the input data, the CRC32 code bit · can finally be obtained. When the end part Ⓓ of the input data is detected, an output selector (MUX) T selects the CRC16 arithmetic operation device ②' S, and the CRC16 ② code bit Ⓓ can finally be obtained from the CRC16 arithmetic operation device ②' S. The CRC16 arithmetic operation device ②' S inputs the end part Ⓓ of the input data, the CRC16 ② arithmetic operation result Ⓒ obtained from the CRC16 arithmetic operation device ② K through the latch L, and the CRC16 ① arithmetic operation result Ⓒ obtained from the CRC16 ① arithmetic operation device D through the latch H, in order to expedite timing.

At this stage, the CRC16 ① arithmetic operation result is obtained by performing the CRC16 ① arithmetic operation using the end part· of the input data and the immediately preceding CRC16 ① arithmetic operation result.

Accordingly, with the above configuration of the second invention having feature in that the CRC16 ① arithmetic operation is included (incorporated) in the CRC16 ② arithmetic operation, it is possible to perform the CRC16 ② arithmetic operation, without using (waiting for) the CRC16 ① arithmetic operation result.

The CRC16 ② arithmetic operation result can be obtained one clock (MIN.) earlier, compared to that in the conventional configuration, since it is not necessary to wait for the CRC16 ① arithmetic operation result. Only one clock delay occurs even on the side of the data path.

In addition, with configuration of the second embodiment, in a stage of the third arithmetic operation performed by the CRC16 arithmetic operation device ③', since the CRC16 ① and CRC16 ② arithmetic operations are included (incorporated) in the CRC16 ③' arithmetic operation, it is possible to perform the CRC16 ② arithmetic operation, without using (waiting for) the CRC16 ① and CRC16 ② arithmetic operation results.

With the circuit configuration of the second embosiment, a time delay being equivalent to one clock occurs between inputting of data and outputting of data. This means reduction of latency by two clocks (MIN.), compared to the conventional circuit configuration.

7

[operational expression producing method]
The CRC arithmetic operation devices, which are used in a conventional example and an embodiment according to the present invention, use the following operational expression.

CRC16 arithmetic operation device G, K, Q:
The device G, K, Q utilize the generating circuit described on page 5 of the previous Document and use a set of output data being output from each of the flip-flops when 32 bits of data were shifted.

CRC16 arithmetic operation device S:
The device S uses the operational expression described on page 11 of this Document obtained in accordance with the operational expression producing method described in the previous Document.

CRC16 arithmetic operation device U:
The device U produces newly an operational expression by using the following method:

If the CRC16 ② arithmetic operation is performed after the CRC16 ① and CRC16 ② arithmetic operation results arithmetic operation result was obtained, a time delay being equivalent to two clocks occurs inevitably.
To solve this problem, it is preferable that the CRC16 ① arithmetic operation is simultaneously performed, when the CRC16 ② arithmetic operation is performed, and the CRC16 ① and CRC16 ② arithmetic operation is simultaneously performed, when the CRC16 ③ arithmetic operation is performed, whereby it becomes possible to acquire simultaneously the CRC16 ①, CRC16 ② and CRC16 ③ arithmetic operation results, without using (waiting for) the CRC16 ① and CRC16 ② arithmetic operation results.

Therefore, in order to avoid such a time delay, a new operational expression is produced and used according to the procedures as below:

8

① Operational Expression Production

CRC16 arithmetic operation device ① G; ② K; ③ Q;
　The device G, K, Q utilize the generating circuit described in the previous Document, and a set of output data being output from each of the flip-flops, when 32 bits of data were shifted, are as follows:

	Operational expression (·:Exculsive OR)
C15	X03·X04·X06·X09·X11·X12·X14·X15· / D00·D01·D03·D04·D06·D09·D11·D12·D17·D19·D20·D24·D28
C14	X02·X03·X05·X08·X10·X11·X13·X14· D01·D02·D04·D05·D07·D10·D12·D13·D18·D20·D21·D25·D29
C13	X01·X02·X04·X07·X09·X10·X12·X13·X15· D00·D02·D03·D05·D06·D08·D11·D13·D14·D19·D21·D22·D26·D30
C12	X00·X01·X03·X06·X08·X09·X11·X12·X14· D01·D03·D04·D06·D07·D09·D12·D14·D15·D20·D22·D23·D27·D31
C11	X00·X02·X03·X04·X05·X06·X07·X08·X09·X10·X12·X13·X14·X15· D00·D01·D02·D03·D05·D06·D07·D08·D09·D10·D11·D12·D13·D15·D16·D17·D19·D20·D21· D23
C10	X01·X02·X03·X04·X05·X06·X07·X08·X09·X11·X12·X13·X14·X15· D00·D01·D02·D03·D04·D06·D07·D08·D09·D10·D11·D12·D13·D14·D16·D17·D18·D20·D21· D22·D24
C09	X00·X01·X02·X03·X04·X05·X06·X07·X08·X10·X11·X12·X13·X14·X15· D00·D01·D02·D03·D04·D05·D07·D08·D09·D10·D11·D12·D13·D14·D15·D17·D18·D19·D21· D22·D23·D25
C08	X00·X01·X02·X03·X04·X05·X06·X07·X09·X10·X11·X12·X13·X14· D01·D02·D03·D04·D05·D06·D08·D09·D10·D11·D12·D13·D14·D15·D16·D18·D19·D20·D22· D23·D24·D26
C07	X00·X01·X02·X03·X04·X05·X06·X08·X09·X10·X11·X12·X13· D02·D03·D04·D05·D06·D07·D09·D10·D11·D12·D13·D14·D15·D16·D17·D19·D20·D21·D23· D24·D25·D27
C06	X00·X01·X02·X03·X04·X05·X07·X08·X09·X10·X11·X12· D03·D04·D05·D06·D07·D08·D10·D11·D12·D13·D14·D15·D16·D17·D18·D20·D21·D22·D24· D25·D26·D28
C05	X00·X01·X02·X03·X04·X06·X07·X08·X09·X10·X11· D04·D05·D06·D07·D08·D09·D11·D12·D13·D14·D15·D16·D17·D18·D19·D21·D22·D23·D25· D26·D27·D29
C04	X00·X01·X02·X03·X05·X06·X07·X08·X09·X10·X15· D00·D05·D06·D07·D08·D09·D10·D12·D13·D14·D15·D16·D17·D18·D19·D20·D22·D23·D24· D26·D27·D28·D30
C03	X00·X01·X02·X04·X05·X06·X07·X08·X09·X14·X15· D00·D01·D06·D07·D08·D09·D10·D11·D13·D14·D15·D16·D17·D18·D19·D20·D21·D23·D24· D25·D27·D28·D29·D31
C02	X00·X01·X05·X07·X08·X09·X11·X12·X13·X15· D00·D02·D03·D04·D06·D07·D08·D10·D14·D15·D16·D18·D21·D22·D25·D26·D29·D30
C01	X00·X04·X06·X07·X08·X10·X11·X12·X14·X15· D00·D01·D03·D04·D05·D07·D08·D09·D11·D15·D16·D17·D19·D22·D23·D26·D27·D30·D31
C00	X04·X05·X07·X10·X12·X13·X15· D00·D02·D03·D05·D08·D10·D11·D16·D18·D19·D23·D27·D31

"X" denotes an initial value of latch (flip-flop) H.

9

② RC16 arithmetic operation device ②′ S:
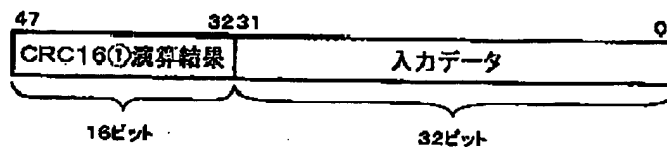   Operational Expression Production

The device S uses the operational expression described
in this Document obtained in accordance with the operational
expression producing method described in the previous Document.

a. Data being of 48 bits in length.

In the CRC16 arithmetic operation device ②′ S, the
operational expression is produced using input data (32 bits)
and an immediately preceding arithmetic operation result (16
bits).

With the conventional method, CRC16 code bit is acquired
by adding the CRC16 code bit (result obtained from CRC16
arithmetic operation device ①: 16 bits) to an end part of the
input data.

The CRC16 arithmetic operation device ②′ S incorporated
in the present invention inputs simultaneously the input data,
and CRC16 ① arithmetic operation result (obtained one operation
before a final CRC16 arithmetic operation result). That is, the
operational expression is produced as 48 bits of the input data.
At this stage, original input data · as lower-order bits and
the CRC arithmetic operation result as higher-order bits (see
below) are respectively input.

47	3231	0
CRC16①演算結果	入力データ	

16ビット   32ビット

b. Operational Expression Production-I

First, a CRC16 operational expression on input data being of 48 bits in length is produced. This is the output data from each of the flip-flops making up the CRC16 generating circuit described on Page 6 in the previous document, when 48 bits of data is shifted.

	operational expressions (·:Excusive OR)
C15	Z01·Z04·Z08·Z10·Z11·Z12·Z13·D02·D03·D04·D05·D07·D11·D14·D16·D17·D19·D20·D22·D25·D27·D28·D33·D35·D36·D40·D44
C14	Z00·Z03·Z07·Z09·Z10·Z11·Z12·D03·D04·D05·D06·D08·D12·D15·D17·D18·D20·D21·D23·D26·D28·D29·D34·D36·D37·D41·D45
C13	Z02·Z06·Z08·Z09·Z10·Z11·Z15·D00·D04·D05·D06·D07·D09·D13·D16·D18·D19·D21·D22·D24·D27·D29·D30·D35·D37·D38·D42·D46
C12	Z01·Z05·Z07·Z08·Z09·Z10·Z14·D01·D05·D06·D07·D08·D10·D14·D17·D19·D20·D22·D23·D25·D28·D30·D31·D36·D38·D39·D43·D47
C11	Z00·Z01·Z06·Z07·Z09·Z10·Z11·Z12·Z15·D00·D03·D04·D05·D06·D08·D09·D14·D15·D16·D17·D18·D19·D21·D22·D23·D24·D25·D26·D27·D28·D29·D31·D32·D33·D35·D36·D37·D39
C10	Z00·Z05·Z06·Z08·Z09·Z10·Z11·Z14·Z15·D00·D01·D04·D05·D06·D07·D09·D10·D15·D16·D17·D18·D19·D20·D22·D23·D24·D25·D26·D27·D28·D29·D30·D32·D33·D34·D37·D36·D38·D40
C09	Z04·Z05·Z07·Z08·Z09·Z10·Z13·Z14·Z15·D00·D01·D02·D05·D06·D07·D08·D10·D11·D16·D17·D18·D19·D20·D21·D23·D24·D25·D26·D27·D28·D29·D30·D31·D33·D34·D35·D37·D38·D39·D41
C08	Z03·Z04·Z06·Z07·Z08·Z09·Z12·Z13·Z14·D01·D02·D03·D06·D07·D08·D09·D11·D12·D17·D18·D19·D20·D21·D22·D24·D25·D26·D27·D28·D29·D30·D31·D32·D34·D35·D36·D38·D39·D40·D42
C07	Z02·Z03·Z05·Z06·Z07·Z08·Z11·Z12·Z13·D02·D03·D04·D07·D08·D09·D10·D12·D13·D18·D19·D20·D21·D22·D23·D25·D26·D27·D28·D29·D30·D31·D32·D33·D35·D36·D37·D39·D40·D41·D43
C06	Z01·Z02·Z04·Z05·Z06·Z07·Z10·Z11·Z12·D03·D04·D05·D08·D09·D10·D11·D13·D14·D19·D20·D21·D22·D23·D24·D26·D27·D28·D29·D30·D31·D32·D33·D34·D36·D37·D38·D40·D41·D42·D44
C05	Z00·Z01·Z03·Z04·Z05·Z06·Z09·Z10·Z11·D04·D05·D06·D09·D10·D11·D12·D14·D15·D20·D21·D22·D23·D24·D25·D27·D28·D29·D30·D31·D32·D33·D34·D35·D37·D38·D39·D41·D42·D43·D45
C04	Z00·Z02·Z03·Z04·Z05·Z08·Z09·Z10·Z15·D00·D05·D06·D07·D10·D11·D12·D13·D15·D16·D21·D22·D23·D24·D25·D26·D28·D29·D30·D31·D32·D33·D34·D35·D36·D38·D39·D40·D42·D43·D44·D46
C03	Z01·Z02·Z03·Z04·Z07·Z08·Z09·Z14·D01·D06·D07·D08·D11·D12·D13·D14·D16·D17·D22·D23·D24·D25·D26·D27·D29·D30·D31·D32·D33·D34·D35·D36·D37·D39·D40·D41·D43·D44·D45·D47
C02	Z00·Z02·Z03·Z04·Z06·Z07·Z10·Z11·Z12·D03·D04·D05·D08·D09·D11·D12·D13·D15·D16·D18·D19·D20·D22·D23·D24·D26·D30·D31·D32·D34·D37·D38·D41·D42·D45·D46·
C01	Z01·Z02·Z03·Z05·Z06·Z09·Z10·Z11·Z15·D00·D04·D05·D06·D09·D10·D12·D13·D14·D16·D17·D19·D20·D21·D23·D24·D25·D27·D31·D32·D33·D35·D38·D39·D42·D43·D46·D47
C00	Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·D01·D02·D03·D04·D06·D10·D13·D15·D16·D18·D19·D21·D24·D26·D27·D32·D34·D35·D39·D43·D47

"Z" denotes an initial value of latch (flip-flop) L.

c. Replacement of data *Page 8*
Operational expressions described earlier are substituted into operational expressions obtained in "b", since D47-D31 are the CRC16 ① arithmetic operation results, as clear from "a".
Replacement example in the least significant bit (C0)

$$C0 = Z00 \cdot Z02 \cdot Z05 \cdot Z09 \cdot Z11 \cdot Z12 \cdot Z13 \cdot Z14 \cdot$$
$$D01 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D10 \cdot D13 \cdot D15 \cdot D16 \cdot D18 \cdot D19 \cdot D21 \cdot D24 \cdot D26 \cdot D27 \cdot D32 \cdot D34 \cdot$$
$$D35 \cdot D39 \cdot D43 \cdot D47$$

P7

$D47 \Leftarrow C15 = X03 \cdot X04 \cdot X06 \cdot X09 \cdot X11 \cdot X12 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D03 \cdot D04 \cdot D06 \cdot D09 \cdot D11 \cdot D12 \cdot D17 \cdot D19 \cdot D20 \cdot D24 \cdot D28$

$D43 \Leftarrow C11 = X00 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X10 \cdot X12 \cdot X13 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D02 \cdot D03 \cdot D05 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D15 \cdot D16 \cdot$
$D17 \cdot D19 \cdot D20 \cdot D21 \cdot D23$

$D39 \Leftarrow C07 = X00 \cdot X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X08 \cdot X09 \cdot X10 \cdot X11 \cdot X12 \cdot X13 \cdot$
$D02 \cdot D03 \cdot D04 \cdot D05 \cdot D06 \cdot D07 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D17 \cdot$
$D19 \cdot D20 \cdot D21 \cdot D23 \cdot D24 \cdot D25 \cdot D27$

$D35 \Leftarrow C03 = X00 \cdot X01 \cdot X02 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D17 \cdot D18 \cdot D19 \cdot$
$D20 \cdot D21 \cdot D23 \cdot D24 \cdot D25 \cdot D27 \cdot D28 \cdot D29 \cdot D31$

$D34 \Leftarrow C02 = X00 \cdot X01 \cdot X05 \cdot X07 \cdot X08 \cdot X09 \cdot X11 \cdot X12 \cdot X13 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D08 \cdot D10 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot D21 \cdot D22 \cdot D25 \cdot$
$D26 \cdot D29 \cdot D30$

$D32 \Leftarrow C00 = X04 \cdot X05 \cdot X07 \cdot X10 \cdot X12 \cdot X13 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D03 \cdot D05 \cdot D08 \cdot D10 \cdot D11 \cdot D16 \cdot D18 \cdot D19 \cdot D23 \cdot D27 \cdot D31$

Substituting

$$C0 = Z00 \cdot Z02 \cdot Z05 \cdot Z09 \cdot Z11 \cdot Z12 \cdot Z13 \cdot Z14 \cdot$$
$$D01 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D10 \cdot D13 \cdot D15 \cdot D16 \cdot D18 \cdot D19 \cdot D21 \cdot D24 \cdot D26 \cdot D27 \cdot$$
$$X03 \cdot X04 \cdot X06 \cdot X09 \cdot X11 \cdot X12 \cdot X14 \cdot X15 \cdot$$
$$D00 \cdot D01 \cdot D03 \cdot D04 \cdot D06 \cdot D09 \cdot D11 \cdot D12 \cdot D17 \cdot D19 \cdot D20 \cdot D24 \cdot D28 \cdot$$
$$X00 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X10 \cdot X12 \cdot X13 \cdot X14 \cdot X15 \cdot$$
$$D00 \cdot D01 \cdot D02 \cdot D03 \cdot D05 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D15 \cdot D16 \cdot D17 \cdot$$
$$D19 \cdot D20 \cdot D21 \cdot D23 \cdot$$
$$X00 \cdot X01 \cdot X02 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X14 \cdot X15 \cdot$$
$$D00 \cdot D01 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D17 \cdot D18 \cdot D19 \cdot$$
$$D20 \cdot D21 \cdot D23 \cdot D24 \cdot D25 \cdot D27 \cdot D28 \cdot D29 \cdot D31 \cdot$$
$$X00 \cdot X01 \cdot X05 \cdot X07 \cdot X08 \cdot X09 \cdot X11 \cdot X12 \cdot X13 \cdot X15 \cdot$$
$$D00 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D08 \cdot D10 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot D21 \cdot D22 \cdot D25 \cdot$$
$$D28 \cdot D29 \cdot D30 \cdot$$
$$X04 \cdot X05 \cdot X07 \cdot X10 \cdot X12 \cdot X13 \cdot X15 \cdot$$
$$D00 \cdot D02 \cdot D03 \cdot D05 \cdot D08 \cdot D10 \cdot D11 \cdot D16 \cdot D18 \cdot D19 \cdot D23 \cdot D27 \cdot D31$$

Deleting same terms

$$C0 = Z00 \cdot Z02 \cdot Z05 \cdot Z09 \cdot Z11 \cdot Z12 \cdot Z13 \cdot Z14 \cdot$$
$$X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X09 \cdot X10 \cdot X11 \cdot X12 \cdot X14 \cdot X15 \cdot$$
$$D00 \cdot D02 \cdot D05 \cdot D11 \cdot D12 \cdot D14 \cdot D15 \cdot D21 \cdot D22 \cdot D25 \cdot D30$$

12

d. Operational expressions

It is possible to obtain CRC16 arithmetic operation device ②' by performing the above processing on all the operational expressions.

	Operational expressions
C15	Z01・Z04・Z08・Z10・Z11・Z12・Z13・ X00・X02・X03・X04・X05・X08・X10・X11・X12・X13・X15 D00・D07・D09・D10・D12・D13・D14・D15・D16・D18・D19・D20・D21・D22・D23・D24・D26・D27・D28・ D29・D31
C14	Z00・Z03・Z07・Z09・Z10・Z11・Z12・ X01・X03・X04・X05・X06・X07・X11・X12・X13・X14・ D01・D02・D05・D06・D09・D10・D11・D14・D15・D16・D18・D21・D22・D24・D25・D26・D29・ D30
C13	Z02・Z06・Z08・Z09・Z10・Z11・Z15・ X00・X02・X06・X08・X10・X14・ D00・D01・D04・D06・D15・D16・D19・D27・D30・D31
C12	Z01・Z05・Z07・Z08・Z09・Z10・Z14・ X01・X03・X07・X09・X11・X15・ D00・D01・D04・D05・D07・D10・D12・D19・D21・D22・D25・D31
C11	Z00・Z01・Z06・Z07・Z09・Z10・Z11・Z12・Z15・ X00・X03・X04・X06・X07・X08・X10・X11・X12・ D00・D06・D07・D11・D12・D14・D16・D18・D19・D20・D23・D27・D28・D29
C10	Z00・Z05・Z06・Z08・Z09・Z10・Z11・Z14・Z15・ X01・X04・X05・X07・X08・X09・X11・X12・X13・ D00・D01・D02・D03・D05・D08・D09・D11・D14・D15・D17・D18・D21・D22・D26・D27・D28・D29
C09	Z04・Z05・Z07・Z08・Z09・Z10・Z13・Z14・Z15・ X00・X02・X05・X06・X08・X09・X10・X12・X13・X14・ D00・D03・D08・D09・D11・D13・D15・D16・D17・D21・D22・D25・D26・D27・D28・D30・D31
C08	Z03・Z04・Z06・Z07・Z08・Z09・Z12・Z13・Z14・ X01・X03・X06・X07・X09・X10・X11・X13・X14・X15・ D00・D03・D04・D05・D07・D11・D14・D20・D23・D24・D25・D26・D27・D29・D30・D31
C07	Z02・Z03・Z05・Z06・Z07・Z08・Z11・Z12・Z13・ X02・X05・X08・X11・X13・X14・ D01・D03・D08・D09・D12・D17・D18・D25・D26・D27・D28・D29・D30
C06	Z01・Z02・Z04・Z05・Z06・Z07・Z10・Z11・Z12・ X03・X04・X05・X06・X07・X09・X10・X13・X14・ D01・D02・D03・D04・D06・D12・D13・D14・D19・D24・D26・D28・D29
C05	Z00・Z01・Z03・Z04・Z05・Z06・Z09・Z10・Z11・ X06・X08・X11・X12・X13・X14・ D01・D02・D03・D05・D06・D07・D10・D11・D12・D14・D15・D16・D18・D24・D25・D26・D28
C04	Z00・Z02・Z03・Z04・Z05・Z08・Z09・Z10・Z15・ X00・X07・X09・X12・X13・X14・X15・ D01・D02・D03・D05・D07・D08・D10・D11・D12・D13・D16・D17・D19・D20・D23・D24・D31
C03	Z01・Z02・Z03・Z04・Z07・Z08・Z09・Z14・ X01・X08・X10・X13・X14・X15・ D00・D02・D05・D06・D08・D11・D12・D13・D17・D18・D19・D20・D21・D22・D23・D26・D28・D30・D31
C02	Z00・Z02・Z03・Z04・Z06・Z07・Z10・Z11・Z12・ X03・X06・X07・X09・X14・X15・ D00・D01・D03・D04・D05・D06・D11・D13・D15・D19・D21・D22・D24・D25・D26・D28・D29・D30
C01	Z01・Z02・Z03・Z05・Z06・Z09・Z10・Z11・Z15・ X00・X05・X08・X12・X13・ D00・D02・D03・D04・D05・D06・D07・D09・D12・D13・D14・D15・D18・D19・D21・D22・D25・D27・D28・ D30,
C00	Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・ X01・X02・X03・X04・X05・X09・X10・X11・X12・X14・X15・ D00・D02・D05・D11・D12・D14・D15・D21・D22・D25・D30

"X" denotes an initial value of latch (flip-flop) H.
"Z" denotes an initial value of latch (flip-flop) L.

13

③ CRC16 arithmetic operation device ③′ U:
   Operational Expression Production

a. Data being of 64 bits in length.
   In the CRC16 arithmetic operation device ③′ S, the operational expression is produced using input data (32 bits) and an immediately preceding arithmetic operation result (16 bits).

With the conventional method, CRC16 code bit is acquired by adding the CRC16 code bit (result obtained from CRC16 arithmetic operation device ②: 16 bits) to an end part of the input data.

The CRC16 arithmetic operation device ③′ S incorporated in the present invention inputs simultaneously the input data, CRC16 ① arithmetic operation result (obtained two operations before a final CRC16 arithmetic operation result), and CRC16 ②′ arithmetic operation result (obtained one operation before a final CRC16 arithmetic operation result). That is, the operational expression is produced as 64 bits of the input data. At this stage, original input data (D) as lower-order bits and the CRC arithmetic operation result as higher-order bits (see below) are respectively input.

63	4847	3231		0
CRC16 ② arithmetic operation result	CRC16 ① arithmetic operation result		Input data	
16bits	16bits		32bits	

14
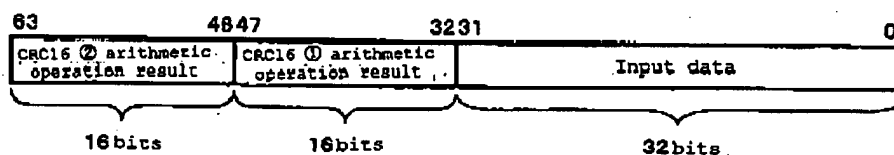
b. Operational Expression Production-I

First, a CRC16 operational expression on input data being of 64 bits in length is produced. This is the output data from each of the flip-flops making up the CRC16 generating circuit described in the previous document, when 64 bits of data is shifted. *on page 6*

	Operational expressions
C15	R01·R03·R04·R07·R08·R10·R11·R12·R13·R15·D00·D02·D03·D04·D05·D07·D08·D11·D12·D14·D18·D19·D20·D21·D23·D27·D30·D32·D33·D35·D36·D38·D41·D43·D44·D49·D51·D52·D56·D60
C14	R00·R02·R03·R06·R07·R09·R10·R11·R12·R14·R15·D00·D01·D03·D04·D05·D06·D08·D09·D12·D13·D15·D19·D20·D21·D22·D24·D28·D31·D33·D34·D36·D37·D39·D42·D44·D45·D50·D52·D53·D57·D61
C13	R01·R02·R05·R06·R08·R09·R10·R11·R13·R14·R15·D00·D01·D02·D04·D05·D06·D07·D09·D10·D13·D14·D16·D20·D21·D22·D23·D25·D29·D32·D34·D35·D37·D38·D40·D43·D45·D46·D51·D53·D54·D58·D62
C12	R00·R01·R04·R05·R07·R08·R09·R10·R12·R13·R14·R15·D00·D01·D02·D03·D05·D06·D07·D08·D10·D11·D14·D15·D17·D21·D22·D23·D24·D26·D30·D33·D35·D36·D38·D39·D41·D44·D46·D47·D52·D54·D55·D59·D63
C11	R00·R01·R06·R09·R10·R14·D01·D05·D06·D09·D14·D15·D16·D19·D20·D21·D22·D24·D30·D31·D32·D33·D34·D35·D37·D38·D39·D40·D41·D42·D43·D44·D45·D47·D48·D49·D51·D52·D53·D55
C10	R00·R05·R08·R09·R13·D02·D06·D07·D10·D15·D16·D17·D20·D21·D22·D23·D25·D26·D31·D32·D33·D34·D35·D36·D38·D39·D40·D41·D42·D43·D44·D45·D46·D48·D49·D50·D52·D53·D54·D56
C09	R04·R07·R08·R12·R15·D00·D03·D07·D08·D11·D16·D17·D18·D21·D22·D23·D24·D26·D27·D32·D33·D34·D35·D36·D37·D39·D40·D41·D42·D43·D44·D45·D46·D47·D49·D50·D51·D53·D54·D55·D57
C08	R03·R06·R07·R11·R14·R15·D00·D01·D04·D08·D09·D12·D17·D18·D19·D22·D23·D24·D25·D27·D28·D33·D34·D35·D36·D37·D38·D40·D41·D42·D43·D44·D45·D46·D47·D48·D50·D51·D52·D54·D55·D56·D58
C07	R02·R05·R06·R10·R13·R14·R15·D00·D01·D02·D05·D09·D10·D13·D18·D19·D20·D23·D24·D25·D26·D28·D29·D34·D35·D36·D37·D38·D39·D41·D42·D43·D44·D45·D46·D47·D48·D49·D51·D52·D53·D55·D56·D57·D59
C06	R01·R04·R05·R09·R12·R13·R14·D01·D02·D03·D06·D10·D11·D14·D19·D20·D21·D24·D25·D26·D27·D29·D30·D35·D36·D37·D38·D39·D40·D42·D43·D44·D45·D46·D47·D48·D49·D50·D52·D53·D54·D56·D57·D58·D60
C05	R00·R03·R04·R08·R11·R12·R13·D02·D03·D04·D07·D11·D12·D15·D20·D21·D22·D25·D26·D27·D28·D30·D31·D36·D37·D38·D39·D40·D41·D43·D44·D45·D46·D47·D48·D49·D50·D51·D53·D54·D55·D57·D58·D59·D61
C04	R02·R03·R07·R10·R11·R12·R15·D00·D03·D04·D05·D08·D12·D13·D16·D21·D22·D23·D26·D27·D28·D29·D31·D32·D37·D38·D39·D40·D41·D42·D44·D45·D46·D47·D48·D49·D50·D51·D52·D54·D55·D56·D58·D59·D60·D62
C03	R01·R02·R06·R09·R10·R11·R14·D01·D04·D05·D06·D09·D13·D14·D17·D22·D23·D24·D27·D28·D29·D30·D32·D33·D38·D39·D40·D41·D42·D43·D45·D46·D47·D48·D49·D50·D51·D52·D53·D55·D56·D57·D59·D60·D61·D63
C02	R00·R03·R04·R05·R07·R09·R11·R12·D03·D04·D06·D08·D10·D11·D12·D15·D19·D20·D21·D24·D25·D27·D28·D29·D31·D32·D34·D35·D36·D38·D39·D40·D42·D46·D47·D48·D50·D53·D54·D57·D58·D61·D62
C01	R02·R03·R04·R06·R08·R10·R11·R15·D00·D04·D05·D07·D09·D11·D12·D13·D18·D20·D21·D22·D25·D26·D28·D29·D30·D32·D33·D35·D36·D37·D39·D40·D41·D43·D47·D48·D49·D51·D54·D55·D58·D59·D62·D63
C00	R02·R04·R05·R08·R09·R11·R12·R13·R14·D01·D02·D03·D04·D06·D07·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·D32·D34·D35·D37·D40·D42·D43·D48·D50·D51·D55·D59·D63

"R" denotes an initial value of latch (flip-flop) R.

c. Replacement of data   Page 12   page 8
Operational expressions described earlier are substituted into
operational expressions obtained in "b", since D63-D48 are the CRC16
②' arithmetic operation results, as clear from "a".
Operational expressions described earlier are substituted into
operational expressions obtained in "b", since D47-D32 are the CRC16
① arithmetic operation results, as clear from "a".
Replacement example in the least significant bit (CO)

$$C0 = R02 \cdot R04 \cdot R05 \cdot R08 \cdot R09 \cdot R11 \cdot R12 \cdot R13 \cdot R14 \cdot$$
$$D01 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D10 \cdot D11 \cdot D13 \cdot D17 \cdot D18 \cdot D19 \cdot D20 \cdot D22 \cdot D26 \cdot D29 \cdot D31 \cdot$$
$$D32 \cdot D34 \cdot D35 \cdot D37 \cdot D40 \cdot D42 \cdot D43 \cdot D48 \cdot D50 \cdot D51 \cdot D55 \cdot D59 \cdot D63$$

P11

D63 <=   C15= $Z01 \cdot Z04 \cdot Z08 \cdot Z10 \cdot Z11 \cdot Z12 \cdot Z13 \cdot$
$X00 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X10 \cdot X11 \cdot X12 \cdot X13 \cdot X15 \cdot$
$D00 \cdot D07 \cdot D09 \cdot D10 \cdot D12 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot D19 \cdot D20 \cdot D21 \cdot D22 \cdot D23 \cdot D24 \cdot$
$D26 \cdot D27 \cdot D28 \cdot D29 \cdot D31$

D59 <=   C11= $Z00 \cdot Z01 \cdot Z06 \cdot Z07 \cdot Z09 \cdot Z10 \cdot Z11 \cdot Z12 \cdot Z15 \cdot$
$X00 \cdot X03 \cdot X04 \cdot X06 \cdot X07 \cdot X08 \cdot X10 \cdot X11 \cdot X12 \cdot$
$D00 \cdot D06 \cdot D07 \cdot D11 \cdot D12 \cdot D14 \cdot D16 \cdot D18 \cdot D19 \cdot D20 \cdot D23 \cdot D27 \cdot D28 \cdot D29$

D55 <=   C07= $Z02 \cdot Z03 \cdot Z05 \cdot Z06 \cdot Z07 \cdot Z08 \cdot Z11 \cdot Z12 \cdot Z13 \cdot X02 \cdot X05 \cdot X08 \cdot X11 \cdot X13 \cdot$
X14·

$D01 \cdot D03 \cdot D08 \cdot D09 \cdot D12 \cdot D17 \cdot D18 \cdot D25 \cdot D26 \cdot D27 \cdot D28 \cdot D29 \cdot D30$

D51 <=   C03= $Z01 \cdot Z02 \cdot Z03 \cdot Z04 \cdot Z07 \cdot Z08 \cdot Z09 \cdot Z14 \cdot X01 \cdot X08 \cdot X10 \cdot X13 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D05 \cdot D06 \cdot D08 \cdot D11 \cdot D12 \cdot D13 \cdot D17 \cdot D18 \cdot D19 \cdot D20 \cdot D21 \cdot D22 \cdot D23 \cdot D26 \cdot$
$D28 \cdot D30 \cdot D31$

D50 <=   C02= $Z00 \cdot Z02 \cdot Z03 \cdot Z04 \cdot Z06 \cdot Z07 \cdot Z10 \cdot Z11 \cdot Z12 \cdot X03 \cdot X06 \cdot X07 \cdot X09 \cdot X14 \cdot$
X15·

$D00 \cdot D01 \cdot D03 \cdot D04 \cdot D05 \cdot D06 \cdot D11 \cdot D13 \cdot D15 \cdot D19 \cdot D21 \cdot D22 \cdot D24 \cdot D25 \cdot D26 \cdot D28 \cdot$
$D29 \cdot D30$

D48 <=   C00= $Z00 \cdot Z02 \cdot Z05 \cdot Z09 \cdot Z11 \cdot Z12 \cdot Z13 \cdot Z14 \cdot$
$X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X09 \cdot X10 \cdot X11 \cdot X12 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D05 \cdot D11 \cdot D12 \cdot D14 \cdot D15 \cdot D21 \cdot D22 \cdot D25 \cdot D30$

P7

D43 <=   C11= $X00 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X10 \cdot X12 \cdot X13 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D02 \cdot D03 \cdot D05 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D15 \cdot D16 \cdot D17 \cdot$
$D19 \cdot D20 \cdot D21 \cdot D23$

D42 <=   C10= $X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X11 \cdot X12 \cdot X13 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D14 \cdot D16 \cdot D17 \cdot$
$D18 \cdot D20 \cdot D21 \cdot D22 \cdot D24$

D40 <=   C08= $X00 \cdot X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X09 \cdot X10 \cdot X11 \cdot X12 \cdot X13 \cdot X14 \cdot$
$D01 \cdot D02 \cdot D03 \cdot D04 \cdot D05 \cdot D06 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot$
$D19 \cdot D20 \cdot D22 \cdot D23 \cdot D24 \cdot D26$

D37 <=   C05= $X00 \cdot X01 \cdot X02 \cdot X03 \cdot X04 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X10 \cdot X11 \cdot$
$D04 \cdot D05 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D11 \cdot D12 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D17 \cdot D18 \cdot D19 \cdot D21 \cdot$
$D22 \cdot D23 \cdot D25 \cdot D26 \cdot D27 \cdot D29$

D35 <=   C03= $X00 \cdot X01 \cdot X02 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D17 \cdot D18 \cdot D19 \cdot D20 \cdot$
$D21 \cdot D23 \cdot D24 \cdot D25 \cdot D27 \cdot D28 \cdot D29 \cdot D31$

D34 <=   C02= $X00 \cdot X01 \cdot X05 \cdot X07 \cdot X08 \cdot X09 \cdot X11 \cdot X12 \cdot X13 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D08 \cdot D10 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot D21 \cdot D22 \cdot D25 \cdot D26 \cdot$
$D29 \cdot D30$

D32 <=   C00= $X04 \cdot X05 \cdot X07 \cdot X10 \cdot X12 \cdot X13 \cdot X15$
$D00 \cdot D02 \cdot D03 \cdot D05 \cdot D08 \cdot D10 \cdot D11 \cdot D16 \cdot D18 \cdot D19 \cdot D23 \cdot D27 \cdot D31$

16

Substituting
CO= Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·

CO= R02·R04·R05·R08·R09·R11·R12·R13·R14·
D01·D02·D03·D04·D06·D07·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·
Z01·Z04·Z08·Z10·Z11·Z12·Z13·
X00·X02·X03·X04·X05·X06·X10·X11·X12·X13·X15·
D00·D07·D09·D10·D12·D13·D14·D15·D16·D18·D19·D20·D21·D22·D23·D24·
D26·D27·D28·D29·D31·
Z00·Z01·Z06·Z07·Z09·Z10·Z11·Z12·Z15·
X00·X03·X04·X06·X07·X08·X10·X11·X12·
D00·D06·D07·D11·D12·D14·D16·D18·D19·D20·D23·D27·D28·D29·
Z02·Z03·Z05·Z06·Z07·Z08·Z11·Z12·Z13·X02·X05·X08·X11·X13·X14·
D01·D03·D08·D09·D12·D17·D18·D25·D26·D27·D28·D29·D30·
Z01·Z02·Z03·Z04·Z07·Z08·Z09·Z14·X01·X08·X10·X13·X14·X15·
D00·D02·D05·D06·D08·D11·D12·D13·D17·D18·D19·D20·D21·D22·D23·D26·
D28·D30·D31·
Z00·Z02·Z03·Z04·Z06·Z07·Z10·Z11·Z12·X03·X06·X07·X09·X14·X15·
D00·D01·D03·D04·D05·D06·D11·D13·D15·D19·D21·D22·D24·D25·D26·D28·
D29·D30·
Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·
X01·X02·X03·X04·X05·X09·X10·X11·X12·X14·X15·
D00·D02·D05·D11·D12·D14·D15·D21·D22·D25·D30·
X00·X02·X03·X04·X05·X06·X07·X08·X09·X10·X12·X13·X14·X15·
D00·D01·D02·D03·D05·D06·D07·D08·D09·D10·D11·D12·D13·D15·D16·D17·
D19·D20·D21·D23·
X01·X02·X03·X04·X05·X06·X07·X08·X09·X11·X12·X13·X14·X15·
D00·D01·D02·D03·D04·D06·D07·D08·D09·D10·D11·D12·D13·D14·D16·D17·
D18·D20·D21·D22·D24·
X00·X01·X02·X03·X04·X05·X06·X07·X09·X10·X11·X12·X13·X14·
D01·D02·D03·D04·D05·D06·D08·D09·D10·D11·D12·D13·D14·D15·D16·D18·
D19·D20·D22·D23·D24·D26·
X00·X01·X02·X03·X04·X06·X07·X08·X09·X10·X11·
D04·D05·D06·D07·D08·D09·D11·D12·D13·D14·D15·D16·D17·D18·D19·D21·
D22·D23·D25·D26·D27·D29·
X00·X01·X02·X04·X05·X06·X07·X08·X09·X14·X15·
D00·D01·D06·D07·D08·D09·D10·D11·D13·D14·D15·D16·D17·D18·D19·D20·
D21·D23·D24·D25·D27·D28·D29·D31·
X00·X01·X05·X07·X08·X09·X11·X12·X13·X15·
D00·D02·D03·D04·D06·D07·D08·D10·D14·D15·D16·D18·D21·D22·D25·D26·
D29·D30·
X04·X05·X07·X10·X12·X13·X15
D00·D02·D03·D05·D08·D10·D11·D16·D18·D19·D23·D27·D31

Deleting same terms
CO= R02·R04·R05·R08·R09·R11·R12·R13·R14·
Z02·Z03·Z04·Z07·Z09·Z10·Z11·
X02·X04·X05·X06·X08·X12·X13·
·D01·D05·D08·D09·D11·D12·D13·D16·D17·D18·D21·D22·D24·D30·D31

d. Operational expressions

It is possible to obtain CRC16 arithmetic operation device ③′ by performing the above processing on all the operational expressions.

	Operational expressions
C15	R01·R03·R04·R07·R08·R10·R11·R12·R13·R15·Z03·Z04·Z05·Z08·Z10·Z11·Z12·X01·X05·X06·X07·X09·X14·D00·D01·D02·D06·D09·D16·D19·D20·D23·D24·D26·D28----
C14	R00·R02·R03·R06·R07·R09·R10·R11·R12·R14·R15·Z00·Z04·Z05·Z06·Z09·Z11·Z12·Z13·X00·X03·X05·X08·X09·X11·X12·X13·X14·D00·D03·D04·D05·D06·D07·D08·D11·D13·D15·D17·D18·D19·D24·D27·D28·D29·D30
C13	R01·R02·R05·R06·R08·R09·R10·R11·R13·R14·R15·Z00·Z01·Z02·Z06·Z07·Z09·Z10·Z11·X01·X04·X05·X06·X07·X08·X11·X12·X13·X15·D01·D03·D04·D09·D11·D14·D15·D18·D20·D21·D22·D23·D31
C12	R00·R01·R04·R05·R07·R08·R09·R10·R12·R13·R14·R15·Z00·Z01·Z02·Z03·Z04·Z05·Z08·Z11·Z13·Z15·X00·X03·X04·X05·X06·X07·X10·X11·X12·X14·D06·D09·D10·D11·D13·D15·D21·D22·D24·D26·D31
C11	R00·R01·R06·R09·R10·R14·Z00·Z01·Z04·Z06·Z07·Z08·Z15·X03·X04·X05·X06·X07·X11·X13·X14·X15·D02·D04·D05·D06·D07·D09·D10·D12·D16·D20·D21·D22·D23·D24·D25·D26·D28·D31
C10	R00·R05·R08·R09·R13·Z01·Z02·Z04·Z08·Z09·Z10·Z12·Z13·Z15·X02·X05·X09·X10·X11·X13·D00·D02·D03·D04·D06·D11·D14·D15·D16·D18·D19·D21·D22·D24·D26·D28·D31
C09	R04·R07·R08·R12·R15·Z02·Z03·Z05·Z09·Z10·Z11·Z13·Z14·X01·X04·X08·X09·X10·X12·D00·D01·D02·D04·D08·D10·D12·D13·D14·D16·D17·D19·D25·D26·D27·D29·D30
C08	R03·R06·R07·R11·R14·R15·Z03·Z04·Z06·Z10·Z11·Z12·Z14·Z15·X00·X04·X06·X07·X08·X12·X14·X15·D00·D01·D05·D07·D09·D15·D18·D21·D22·D24·D25·D26·D27·D29·D30
C07	R02·R05·R06·R10·R13·R14·R15·Z00·Z02·Z05·Z09·Z10·Z12·Z13·Z14·X01·X02·X03·X07·X10·X11·X13·X14·D00·D01·D02·D03·D04·D05·D06·D08·D09·D12·D13·D14·D15·D17·D19·D26·D28·D30
C06	R01·R04·R05·R09·R12·R13·R14·Z00·Z01·Z02·Z03·Z04·Z06·Z07·Z09·Z13·X00·X03·X04·X05·X06·X11·X13·X14·X15·D00·D02·D03·D04·D08·D11·D13·D16·D17·D20·D21·D22·D23·D24·D25
C05	R00·R03·R04·R08·R11·R12·R13·Z00·Z01·Z03·Z04·Z07·Z08·Z09·Z10·Z11·Z12·Z13·X01·X09·X11·X13·X15·D00·D02·D04·D05·D08·D17·D20·D23·D26·D27·D28·D29·D31
C04	R02·R03·R07·R10·R11·R12·R15·Z01·Z02·Z07·Z08·Z09·Z11·Z14·Z15·X00·X03·X04·X06·X08·X09·X10·X11·D01·D03·D04·D09·D11·D14·D15·D18·D20·D21·D22·D23·D31
C03	R01·R02·R06·R09·R10·R11·R14·Z00·Z02·Z03·Z04·Z05·Z07·Z08·Z09·Z13·X02·X04·X05·X06·X07·X08·X10·X11·X12·X14·X15·D00·D02·D03·D12·D13·D14·D15·D16·D24·D26·D27·D28·D29·D30·D31
C02	R00·R03·R04·R05·R07·R09·R11·R12·Z00·Z01·Z02·Z04·Z06·Z07·Z10·Z12·Z15·X01·X02·X05·X09·X12·X13·X14·D00·D01·D02·D03·D04·D05·D09·D12·D18·D20·D23·D25·D30·D31
C01	R02·R03·R04·R06·R08·R10·R11·R15·Z01·Z03·Z07·Z08·Z09·Z12·Z14·X00·X02·X03·X05·X08·X09·X10·X13·X14·X15·D02·D03·D04·D07·D08·D09·D10·D11·D12·D14·D15·D17·D18·D19·D25·D26·D27·D28·D29·D31·
C00	R02·R04·R05·R08·R09·R11·R12·R13·R14·Z02·Z03·Z04·Z07·Z09·Z10·Z11·X02·X04·X05·X06·X08·X12·X13·D01·D05·D08·D09·D11·D12·D13·D16·D17·D18·D21·D22·D24·D30·D31

"R" denotes an initial value of latch (flip-flop) R.
"X" denotes an initial value of latch (flip-flop) H.
"Z" denotes an initial value of latch (flip-flop) L.

Replacement example in the least significant bit (C0)

C0= R2·R4·R5·R8·R9·R11·R12·R13·R14·
   D1·D2·D3·D4·D6·D7·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·
   D32·D34·D35·D37·D40·D42·D43·D48·D50·D51·D55·D59·D63

P7, 8

D63 <= C31= R5·R8·R9·R11·R15·R23·R24·R25·R27·R28·R29·R30·R31·
D0·D1·D2·D3·D4·D6·D7·D8·D16·D20·D22·D23·D26

D59 <= C27= R1·R4·R5·R7·R11·R19·R20·R21·R23·R24·R25·R26·R27·
R29·
D2·D4·D5·D6·D7·D8·D10·D11·D12·D20·D24·D26·D27·D30

D55 <= C23= R0·R1·R6·R9·R13·R15·R16·R17·R19·R20·R26·R27·R29·
R31·
D0·D2·D4·D5·D11·D12·D14·D15·D16·D18·D22·D25·D30·D31

D51 <= C19= R3·R7·R8·R11·R15·R16·R20·R22·R24·R25·R27·R29·
D2·D4·D6·D7·D9·D11·D15·D16·D20·D23·D24·D28

D50 <= C18= R2·R6·R7·R10·R14·R15·R19·R21·R23·R24·R26·R28·R31·
D0·D3·D5·D7·D8·D10·D12·D16·D17·D21·D24·D25·D29

D48 <= C16= R0·R4·R5·R8·R12·R13·R17·R19·R21·R22·R24·R26·R29·
R30·
D1·D2·D5·D7·D9·D10·D12·D14·D18·D19·D23·D26·D27·D31

D43 <= C11= R0·R1·R3·R4·R9·R12·R14·R15·R16·R17·R20·R24·R25·
R26·R27·R28·R31·
D0·D3·D4·D5·D6·D7·D11·D14·D15·D16·D17·D19·D22·D27·D28·D30·
D31

D42 <= C10= R0·R2·R3·R5·R9·R13·R14·R16·R19·R26·R28·R29·R31·
D0·D2·D3·D5·D12·D15·D17·D18·D22·D26·D28·D29·D31

D40 <= C8= R0·R1·R3·R4·R8·R10·R11·R17·R22·R28·R31·
D0·D3·D8·D9·D14·D19·D20·D21·D23·D27·D28·D30·D31

D37 <= C5= R0·R1·R3·R4·R5·R6·R7·R10·R18·R19·R20·R21·R24·R28·R29·
D2·D3·D7·D10·D11·D12·D18·D21·D24·D25·D26·D27·D28·D30·D31

D35 <= C3= R1·R2·R3·R7·R8·R9·R10·R14·R15·R17·R18·R19·R25·R27·R31·
D0·D4·D6·D12·D13·D14·D16·D17·D21·D22·D23·D24·D28·D29·D30

D34 <= C2= R0·R2·R6·R7·R8·R9·R13·R14·R16·R17·R18·R24·R26·R30·R31·
D0·D1·D5·D7·D13·D14·D15·D17·D18·D22·D23·D24·D25·D29·D30·
D31

D32 <= C0= R0·R6·R9·R10·R12·R16·R24·R25·R26·R28·R29·R30·R31·
D0·D1·D2·D3·D5·D6·D7·D15·D19·D21·D22·D25·D31

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Appln. Of: KOTAKA

Serial No.:   10/090,302

Filed: March 4, 2002

For: ARITHMETIC OPERATION METHOD FOR CYCLIC ...

Group: 2133

Examiner: Dipakkumar Gandhi   DOCKET: NEC N01321

MAIL STOP AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 CFR 1.131 OF PRIOR INVENTION IN A WTO MEMBER COUNTRY TO OVERCOME A CITED PATENT REFERENCE

Dear Sir:

The undersigned, being the named inventor of the subject application, declares and states the following:

(1)   I conceived of and completed the invention described and claimed in the subject application, in Japan, prior to January 26, 2001, the U.S. filing date of the Ishiwaki U.S. Patent No. 6,725,415 cited in the Office Action mailed September 24, 2004 in the above matter.

(2)   As proof thereof, I provide the following:

(a) Exhibit A, which is a full and complete copy of a written Invention Disclosure, which I prepared and submitted to the IP Division of NEC Electronics Corporation, the assignee of the subject application. As can be seen, the drawing figures attached to Exhibit A essentially correspond to the drawing figures submitted with the subject application. A verified English translation of Exhibit A also is attached hereto.

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

179 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

(b)   My Invention Disclosure was accepted for filing by NEC Electronics Corporation, and a detailed description was then supplied to an outside Japanese Patent Law Firm, who then prepared the documents for filing a Japanese patent application. The application was prepared, reviewed by me, and filed in the Japanese Patent Office as Japanese Patent Application Serial No. 2001-059807 filed March 5, 2001.

(3) The foregoing and attachments clearly show a date of conception and completion of the invention of this application all prior to the January 26, 2001 U.S. filing date of the Ishiwaki U.S. patent application. Moreover, having conceived of the invention prior to January 26, 2001, I proceeded diligently to prepare a complete written disclosure of same and to then promptly file a patent application, initially in Japan, and thereafter, in the United States, covering the invention. At no time between my conception of the invention, and my filing of the subject U.S. Patent Application, did I ever intend to abandon the invention.

As the named inventor, I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Shigenari Kotaka

Shigenari KOTAKA

Date December 24, 2004

2

SERIAL NO.: 10/090,302
DOCKET: NEC N01321

EXHIBIT A

配布先 西村国際特許事務所 西村　征生　殿 番号：（1頁のみ） 10-22440 知財部　高野マネージャー殿	コンカレント出願 明細書作成依頼書	EDC知財ー特出ー　1330
		平成 12年 12月 11日
		日本電気株式会社 エレクトロンデバイス知的財産部 マネージャー 和田　正

担当：久野　淑己
電話：044-435-1421
FAX：044-435-1871

下記の件について明細書作成を依頼します。

――― 記 ―――

1. 整理番号　　：754-10092

2. 提案名称　　：ＣＲＣ演算回路
   　　概要：infini bandのＣＲＣ演算においてレイテンシを削減するような回路

3. 提案者名　　：小高　重成　　　　（三シス　・一設Ｇ　　　）

4. 当部担当者：久野　淑己

5. 添付書類　　：アイデア提案書
   　　　　　　　打ち合わせ記録
   　　　　　　　特開平
   　　　　　　　特開平

6. 希望納期　　：草稿ー　平成12年01月13日
   　　　　　　　出願ー　平成12年01月20日

7. 打合せ場所：玉川　　地区を予定。

8. 備考：
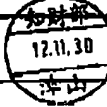
> ┌─ 注記 ─┐
> 　打ち合わせの記録を残し、明細書作成時の一助となるように「コンカレント打ち合わせ記録」を使うようにしました。（原則としてEDC知財部担当者が打ち合わせ時に記入）
> 　この用紙にはEDC知財部と発明者との１ｓｔ打ち合わせの内容がまとめて有り、２ｎｄ打ち合わせの内容もまとめられる様になっていますのでご活用下さい。尚、貴事務所においては１ｓｔ打ち合わせにとらわれることなく、プロの目から見た発明のとらえ方、ストーリの立て方、クレーム案等について事前に検討しておいて下さい。

（第4版：1998.02.25） | 本文第1/6頁

アイデア提案書	事業部整理番号： 754-10092

提案日：2000年 11月 20日 | グループコード： | 部内番号：

[承認欄]部長： ㊞ 課長： ㊞ 主任： | 半特技受付日： 12.11.30

[提案者記入欄]　　　　　　　E-mail: kotaka@lsi.nec.co.jp
提案者所属：第三シスL(事)第1設計　連絡先TEL：822-26011　メール番号：22-26010
提案者氏名：小高重成　　　　社員番号：0690257
適用・応用分野：通信
適用製品名： SUN Wings　　　　売上規模： 3K　　　　（百万円／年）
実験・試作状況：○実験・試作完了 ●実験・試作中 ○実験・試作予定あり ○実験・試作予定なし
先行特許調査(調査した中で近い特許公開番号)：済み、（特開平02-119319）
先行文献調査(調査した中で近い公知例)：済み、無し
特許検索式：(CRC生成回路＋CRC演算回路)&(高速＋高速処理＋レイテンシ＋レイテンシー＋Latency)
関連提案・特許：無し　　　　　　　　　　　　　　1月版
サンプル出荷／社外発表予定：○無 ●有(早い方の日：H13年_9_月_日、何処で　　　　　)
出願希望種別：●コンカレント ○S級 ○通常出願(届出予定日：1999年12月17日)

[上司記入欄]
上司氏名：　堀口立二
実施見込み：●実施決定 ○可能性有り(2年以内)　○可能性有り(4年以内)
　　　　　　○不明 ○見込みなし(理由：　　　　　　　　　　　　　　　)
外国出願希望：○無 ●有(国名：●米 ○韓 ○中国 ○台湾 ○英 ○独 その
他　　　　　)
コメント：

[発明相談コメント欄]　　センター担当：　　　　　年　月　日

[評価委員会記入欄]
評価責任者氏名：　古賀隆俊
決定日：　㊞2000年 11月 27日
評価結果①. 出願希望(●コンカレント ○S級 ○通常届出) 2.公開技報 3.中止 4.再検討
外国出願希望：○無 ●有(国名：●米 ○韓 ○中国 ○台湾 ○英 ○独 その
他　　　　　)
届出指定日：㊞2000年 12月 17日
コメント：計算方法を明確に。米国出願ではアルゴリズムの権利化も検討して
下さい。複数バイトでの処理例を追加し、権利範囲を広げて下さい。
古賀

センターへの要望：

[証人署名欄]
本提案書(図面を含む)の第1ページから第6ページを読んで発明内容を理解しました。
氏名：　堀口立二　　2000年11月20日
[発明者署名欄]
氏名：　小高重成　　2000年11月20日
氏名：　　　　　　　20　年　月　日

【CRC】

　CRC(Cyclic Redundancy Check)とは巡回冗長検査のことでデータ伝送やディスク、テープなどへの読み書きにおいて、データが正しく伝送(読み書き)できたかどうかを検査するための方法の1つです。

　CRC生成多項式と呼ばれる、シフトや加算などを組み合わせた方式で計算を行ないます。CRCの値は通常は16bitか32bitのものが使われます(cyclicとは、2のべき乗を法として、演算結果のオーバーフローを無視するところからきています)。

　CRCの計算は単純な加算だけではないので、ソフトウェアで行なうと負荷が大きいが、ハードウェアで実装すると簡単なため、ディスクコントローラや通信用LSIなどではCRC方式がよく使われています。

<div align="right">—アスキー・デジタル用語辞典から—</div>

【CRC生成多項式】

　上述のCRC生成多項式は、以下のように決められています。

　CRC32(32bit)：$G(X) = X^{32}+X^{20}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1$

　CRC16(16bit)：$G(X) = X^{16}+X^{12}+X^{5}+X^{1}+1$

【CRC方式】

　次に、CRC方式に関して簡単に説明します。なお、上記の32ビットや16ビットでは説明(演算式など)が長くなってしまいますので、ここでは6ビットを例に説明させていただきます。32ビット、16ビットでも考え方は同一です。

　①入力データを数値と見なし多項式で表します。

$$\begin{array}{|c|c|c|c|c|c|}
\hline
1 & 1 & 0 & 0 & 1 & 1 \\
\hline
X^0 & X^1 & X^2 & X^3 & X^4 & X^5
\end{array}$$

$$P(X) = 1 + X^1 + 0 + 0 + X^4 + X^5$$

　②次に送受信であらかじめ決められているCRC生成多項式を用います。

　(CRC32，CRC16では上述の生成多項式)

　　CRC6(6bit)：$G(X) = X^6+X^3+1$

　③G(X)の最高次の項：X^6をP(X)にかけ、この式をQ(X)とします。

　　$Q(X) = X^{11}+X^{10}+X^7+X^6$

④Q（X）をG（X）で割り算します。

ここで、あまり（剰余）がCRC方式のチェック・ビットとなり、CRC符号と呼ばれます。

$$X^5 + X^4 + X^2 + 1$$
$$X^6 + X^3 + 1 \overline{)\, X^{11} + X^{10} \quad + X^7 + X^6}$$
$$\underline{X^{11} \quad + X^8 \quad + X^5}$$
$$X^{10} + X^8 + X^7 + X^6 + X^5$$
$$\underline{X^{10} \quad + X^7 \quad + X^4}$$
$$X^8 \quad + X^6 + X^5 + X^4$$
$$\underline{X^8 \quad + X^5 \quad + X^2}$$
$$X^6 \quad + X^4 \quad + X^2$$
$$\underline{X^6 \quad + X^3 \quad + 1}$$

あまり（CRC符号）＝$X^4 + X^3 + X^2 + 1$

1	0	1	1	1	0

$X^0 \; X^1 \; X^2 \; X^3 \; X^4 \; X^5$

⑤次の入力データに④で得られたCRC符号をかけ、これを新たなQ（X）とします。

このQ（X）をG（X）で割り算することで、新たなCRC符号が得られます。

この繰り返し（巡回）を全データ分行い、転送データの最後に付加して送信します。

【CRC演算式】

CRCの概念は上記の通りです。しかし、割り算器を単純にハード化したのではCRC32など多ビットの場合、高速処理に不向きであることや回路規模が増大してしまうことから、一般的に次の方法を用います。（今回も上記の6ビットを例に説明します。）

まず、前述のCRC生成多項式から、次の回路を得ることができます。

$G(X) = X^6 + X^3 + 1$　　⊗ : Exclusive-OR
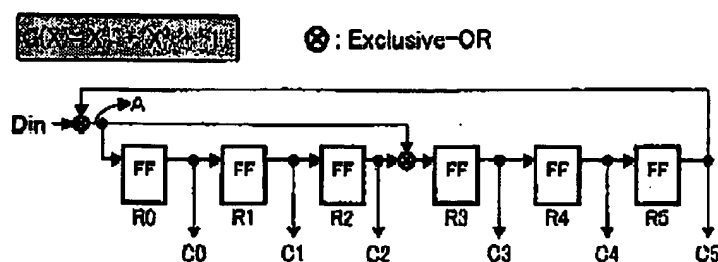
次にCRC方式の項で使用した入力データ（P（X））をDinからシリアルに入力した場合の、各FFの出力状態を以下に示します。

―入力データ―

1	1	0	0	1	1

$X^0 \; X^1 \; X^2 \; X^3 \; X^4 \; X^5$

なお、ループバック・データと入力データのEx-ORをAで表します。

シフト状態	入力値	A	FF出力						備考
			C0	C1	C2	C3	C4	C5	
0	—	—	0	0	0	0	0	0	初期値
1	1	1	1	0	0	1	0	0	
2	1	1	1	1	0	1	1	0	
3	0	0	0	1	1	0	1	1	
4	0	1	1	0	1	0	0	1	
5	1	0	0	1	0	1	0	0	
6	1	1	1	0	1	1	1	0	剰余

上記の剰余とCRC方式で説明した剰余が一致。すなわち、入力データ分シフトさせた際の各FFの出力が求めるべきCRC符号となります。

次に、入力データを$D_0 \sim D_5$、各FFの初期値を$R_0 \sim R_5$、各FFの出力を$C_0 \sim C_5$とすると次のような演算式を導くことができます。

シフト状態	入力値	A	FF出力	
0	—	—	C5	R5
			C4	R4
			C3	R3
			C2	R2
			C1	R1
			C0	R0
1	D5	R5·D5	C5	R4
			C4	R3
			C3	R2·R5·D5
			C2	R1
			C1	R0
			C0	R5·D5
2	D4	R4·D4	C5	R3
			C4	R2·R5·D5
			C3	R1·R4·D4
			C2	R0
			C1	R5·D5
			C0	R4·D4
3	D3	R3·D3	C5	R2·R5·D5
			C4	R1·R4·D4
			C3	R0·R3·D3
			C2	R5·D5
			C1	R4·D4
			C0	R3·D3
4	D2	R2·R5·D5·D2	C5	R1·R4·D4
			C4	R0·R3·D3
			C3	R5·D5·R2·R5·D5·D2
			C2	R4·D4
			C1	R3·D3
			C0	R2·R5·D5·D2

注)・:Ex-ORを意味します。

シフト状態	入力値	A	FF出力	
5	D1	R1·R4·D4·D1	C5	R0·R3·D3
			C4	R5·D5·R2·R5·D5·D2
			C3	R4·D4·R1·R4·D4·D1
			C2	R3·D3
			C1	R2·R5·D5·D2
			C0	R1·R4·D4·D1
6	D0	R0·R3·D3·D0	C5	R5·D5·R2·R5·D5·D2
			C4	R4·D4·R1·R4·D4·D1
			C3	R3·D3·R0·R3·D3·D0
			C2	R2·R5·D5·D2
			C1	R1·R4·D4·D1
			C0	R0·R3·D3·D0

注）・:Ex·ORを意味します。

6シフト目の演算式が、CRC6の演算式となります。ここで、同一項（R3·R3など）は削除可能ですから、上記演算式を整理すると以下が求めるべき演算式となります。

<u>CRC6演算式</u>

$C5 = R2 \cdot D2$
$C4 = R1 \cdot D1$
$C3 = R0 \cdot D0$
$C2 = R2 \cdot R5 \cdot D2 \cdot D5$
$C1 = R1 \cdot R4 \cdot D1 \cdot D4$
$C0 = R0 \cdot R3 \cdot D0 \cdot D3$

上記の式に初期値"0"（R0～R5="0"）、データ（11011）を入力すると、以下のようになります。

$C5 = 0$
$C4 = 1$
$C3 = 1$
$C2 = 1$
$C1 = 0$
$C0 = 1$

これは前述した、結果と一致しています。よって、上記の演算式で問題なしといえます。

以上がCRC6での方式説明と演算式算出までです。

これらと同様にCRC16およびCRC32も演算式を算出できます。以降にCRC16、CRC32の生成多項式および生成回路、演算式を示します。

【CRC16】

・生成多項式

$$G(X) = X^{16} + X^{12} + X^8 + X^1 + 1$$

・生成回路

⊗ : Exclusive-OR

・演算式

C0	R0・R4・R8・R12・R13・R15・D0・D2・D3・D7・D11・D15
C1	R0・R1・R4・R5・R8・R9・R12・R14・R15・D0・D1・D3・D6・D7・D10・D11・D14・D15
C2	R1・R2・R5・R6・R9・R10・R13・R15・D0・D2・D5・D6・D9・D10・D13・D14
C3	R0・R2・R3・R4・R6・R7・R8・R10・R11・R12・R13・R14・R15・D0・D1・D2・D3・D4・D5・D7・D8・D9・D11・D12・D13・D15
C4	R1・R3・R4・R5・R7・R8・R9・R11・R12・R13・R14・R15・D0・D1・D2・D3・D4・D6・D7・D8・D10・D11・D12・D14
C5	R2・R4・R5・R6・R8・R9・R10・R12・R13・R14・R15・D0・D1・D2・D3・D5・D6・D7・D9・D10・D11・D13
C6	R3・R5・R6・R7・R9・R10・R11・R13・R14・R15・D0・D1・D2・D4・D5・D6・D8・D9・D10・D12
C7	R4・R6・R7・R8・R10・R11・R12・R14・R15・D0・D1・D3・D4・D5・D7・D8・D9・D11
C8	R5・R7・R8・R9・R11・R12・R13・R15・D0・D2・D3・D4・D6・D7・D8・D10
C9	R6・R8・R9・R10・R12・R13・R14・D1・D2・D3・D5・D6・D7・D9
C10	R7・R9・R10・R11・R13・R14・R15・D0・D1・D2・D4・D5・D6・D8
C11	R8・R10・R11・R12・R14・R15・D0・D1・D3・D4・D5・D7
C12	R0・R4・R8・R9・R11・D4・D6・D7・D11・D15
C13	R1・R5・R9・R10・R12・D3・D5・D6・D10・D14
C14	R2・R6・R10・R11・R13・D2・D4・D5・D9・D13
C15	R3・R7・R11・R12・R14・D1・D3・D4・D08・D12

なお、上記演算式は入力データ16ビット（D0～D15）の場合です。入力データ幅が他の場合には、別演算式となります。例えば、8ビット（1バイト：D0～D7）時はD7が入力された時点（8シフト）での各FF出力が、必要とされる演算式となります。

【CRC32】

-生成多項式

CRC32（32bit）: $G(X) = X^{32}+X^{20}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^8+X^7+X^5+X^4+X^2+X^1+1$

-生成回路

⊗ : Exclusive-OR

Din →

FF R0 — C0
FF R1 — C1
FF R2 — C2
FF R3 — C3
FF R4 — C4
FF R5 — C5
FF R6 — C6
FF R7 — C7

FF R8 — C8
FF R9 — C9
FF R10 — C10
FF R11 — C11
FF R12 — C12
FF R13 — C13
FF R14 — C14
FF R15 — C15

FF R16 — C16
FF R17 — C17
FF R18 — C18
FF R19 — C19
FF R20 — C20
FF R21 — C21
FF R22 — C22
FF R23 — C23

FF R24 — C24
FF R25 — C25
FF R26 — C26
FF R27 — C27
FF R28 — C28
FF R29 — C29
FF R30 — C30
FF R31 — C31

-演算式

C0	R0·R6·R9·R10·R12·R16·R24·R25·R26·R28·R29·R30·R31·D0·D1·D2· D3·D5·D6·D7·D15·D19·D21·D22·D25·D31
C1	R0·R1·R6·R7·R9·R11·R12·R13·R16·R17·R24·R27·R28·D3·D4·D7· D14·D15·D18·D19·D20·D22·D24·D25·D30·D31
C2	R0·R2·R6·R7·R8·R9·R13·R14·R16·R17·R18·R24·R26·R30·R31·D0· D1·D5·D7·D13·D14·D15·D17·D18·D22·D23·D24·D25·D29·D30·D31
C3	R1·R2·R3·R7·R8·R9·R10·R14·R15·R17·R18·R19·R25·R27·R31·D0· D4·D6·D12·D13·D14·D16·D17·D21·D22·D23·D24·D28·D29·D30
C4	R0·R2·R3·R4·R6·R8·R11·R12·R15·R18·R19·R20·R24·R25·R29·R30· R31·D0·D1·D2·D6·D7·D11·D12·D13·D16·D19·D20·D23·D25·D27· D28·D29·D31
C5	R0·R1·R3·R4·R5·R6·R7·R10·R13·R19·R20·R21·R24·R28·R29·D2· D3·D7·D10·D11·D12·D18·D21·D24·D25·D26·D27·D28·D30·D31
C6	R1·R2·R4·R5·R6·R7·R8·R11·R20·R21·R25·R30·D1·D2·D6·D9·D10· D11·D17·D20·D23·D24·D25·D26·D27·D29·D30
C7	R0·R2·R3·R5·R7·R8·R10·R15·R16·R21·R22·R23·R24·R28·R29·D2· D3·D6·D7·D8·D9·D10·D15·D16·D21·D23·D24·D26·D28·D29·D31
C8	R0·R1·R3·R4·R8·R10·R11·R17·R22·R28·R31·D0·D3·D8·D9·D14· D19·D20·D21·D23·D27·D28·D30·D31
C9	R1·R2·R4·R5·R9·R11·R12·R13·R18·R23·R24·R29·D2·D7·D8·D13· D18·D19·D20·D22·D26·D27·D29·D30
C1 0	R0·R2·R3·R5·R9·R13·R14·R16·R19·R26·R28·R29·R31·D0·D2·D3· D5·D12·D15·D17·D18·D22·D26·D28·D29·D31
C1 1	R0·R1·R3·R4·R9·R12·R14·R15·R16·R17·R20·R24·R25·R26·R27· R28·R31·D0·D3·D4·D5·D6·D7·D11·D14·D15·D16·D17·D19·D22· D27·D28·D30·D31
C1 2	R0·R1·R2·R4·R5·R6·R9·R12·R13·R15·R17·R18·R24·R30·R31·D0· D1·D4·D7·D10·D13·D14·D16·D18·D19·D22·D25·D26·D27·D29·D30· D31
C1 3	R1·R2·R3·R5·R6·R7·R10·R13·R16·R19·R22·R28·R31·D0·D3·D6· D9·D12·D13·D15·D17·D18·D21·D24·D25·D26·D28·D29·D30
C1 4	R02 R03 R04 R06 R07 R08 R11 R14 R15 R17 R19 R20 R23 R26 R29 D02 D05 D08 D11 D12 D14 D16 D17 D20 D23 D24 D25 D27 D28 D29
C1 5	R3·R4·R5·R7·R8·R9·R12·R15·R16·R18·R20·R21·R24·R27·R30·D1· D4·D7·D10·D11·D13·D15·D16·D19·D22·D23·D24·D26·D27·D28
C1 6	R0·R4·R5·R8·R12·R13·R17·R19·R21·R22·R24·R26·R29·R30·D1·D2· D5·D7·D9·D10·D12·D14·D18·D19·D23·D26·D27·D31
C1 7	R1·R5·R6·R9·R13·R14·R18·R20·R22·R25·R27·R30·R31·D0·D1·D4· D6·D8·D9·D11·D13·D17·D18·D22·D25·D26·D30
C1 8	R2·R6·R7·R10·R14·R15·R19·R21·R23·R24·R26·R28·R31·D0·D3·D5· D7·D8·D10·D12·D16·D17·D21·D24·D25·D29
C1 9	R3·R7·R8·R11·R15·R16·R20·R22·R24·R25·R27·R29·D2·D4·D6·D7· D9·D11·D15·D16·D20·D23·D24·D28
C2 0	R4·R8·R9·R12·R16·R17·R21·R23·R25·R26·R28·R30·D1·D3·D5·D6· D8·D10·D14·D15·D19·D22·D23·D27
C2 1	R5·R9·R10·R13·R17·R18·R22·R24·R26·R27·R29·R31·D0·D2·D4·D5· D7·D9·D13·D14·D18·D21·D22·D26
C2 2	R0·R9·R11·R12·R14·R16·R18·R19·R23·R24·R26·R27·R29·R31·D0· D2·D4·D5·D7·D8·D12·D13·D15·D17·D19·D20·D22·D31

C23	R0·R1·R6·R9·R13·R15·R16·R17·R19·R20·R26·R27·R29·R31·D0·D2·D4·D5·D11·D12·D14·D15·D16·D18·D22·D25·D30·D31
C24	R1·R2·R7·R10·R14·R16·R17·R18·R20·R21·R27·R28·R30·D1·D3·D4·D10·D11·D13·D14·D15·D17·D21·D24·D29·D30
C25	R2·R3·R8·R11·R16·R17·R18·R19·R21·R22·R28·R29·R31·D0·D2·D3·D9·D10·D12·D13·D14·D16·D20·D23·D28·D29
C26	R0·R3·R4·R6·R10·R18·R19·R20·R22·R23·R24·R25·R26·R28·R31·D0·D3·D5·D6·D7·D8·D9·D11·D12·D13·D21·D25·D27·D28·D31
C27	R1·R4·R5·R7·R11·R19·R20·R21·R23·R24·R25·R26·R27·R29·D2·D4·D5·D6·D7·D8·D10·D11·D12·D20·D24·D26·D27·D30
C28	R2·R5·R6·R8·R12·R20·R21·R22·R24·R25·R26·R27·R28·R30·D1·D3·D4·D5·D6·D7·D9·D10·D11·D19·D23·D25·D26·D29
C29	R3·R6·R7·R9·R13·R21·R22·R23·R25·R26·R27·R28·R29·R31·D0·D2·D3·D4·D5·D6·D8·D9·D10·D18·D22·D24·D25·D28
C30	R4·R7·R8·R10·R14·R22·R23·R24·R26·R27·R28·R29·R30·D1·D2·D3·D4·D5·D7·D8·D9·D17·D21·D23·D24·D27
C31	R5·R8·R9·R11·R15·R23·R24·R25·R27·R28·R29·R30·R31·D0·D1·D2·D3·D4·D6·D7·D8·D16·D20·D22·D23·D26

なお、上記演算式は入力データ32ビット(4バイト:D0～D31)の場合です。入力データ幅が他の場合には、別演算式となります。例えば、8ビット(1バイト:D0～D7)時はD7が入力された時点(8シフト)での各FF出力が、必要とされる演算式となります。また、64ビット(8バイト:D0～D63)時はD63が入力された時点(64シフト)での各FF出力が、必要とされる演算式となります。

以上までがCRC方式および演算式の一般的な説明です。

以降に本発明に関する補足を行います。

【データ・フォーマット】

　はじめに、一般的なデータ・フォーマットに関して説明します。一般的には以下のようにヘッダー情報とデータ、およびCRCから構成されます。（システムによってはCRCを用いずに、他のエラー検出方法を使用する場合もありますが、今回は本特許とは無関係ですので省かせていただきます。）システムによりCRCは32ビット、16ビットのいずれか（当然他の場合も存在します）を使用します。本特許ではこのCRCが2つ（以上）必要とされるようなシステム（InfiniBandなど）に有効となります。

Header	Data	CRC32	CRC16

CRC32

CRC16

例) 4バイト（32ビット）転送

1byte

#1	data0	data1	data2	data3
#2	data4	data5	data6	data7
#3	data8	data9	data10	data11

#n-2	data n-5	data n-4	data n-3	data n-2
#n-1	data n-1	data n	CRC32	CRC32
#n	CRC32	CRC32	CRC16	CRC16
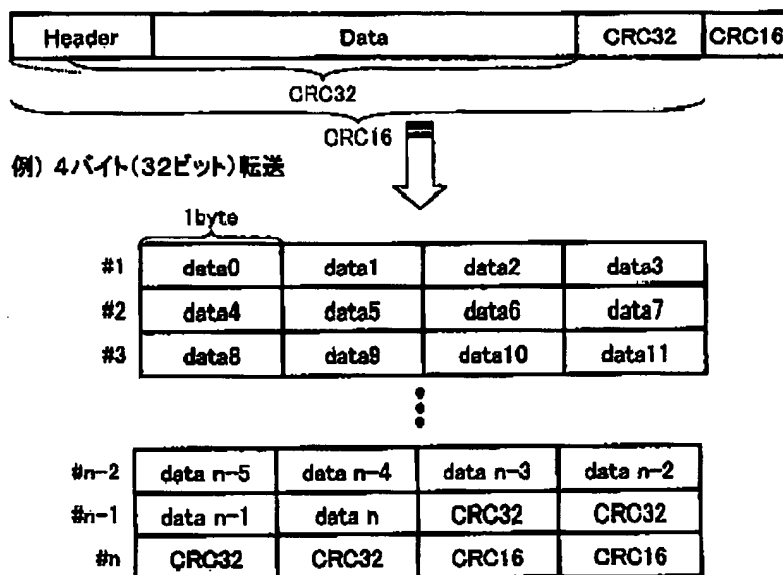
　CRCは転送データのエラーを検出するためのものです。つまり、上記のようなデータを転送する場合、data0～data nまでのエラー検出に使用されます。CRC16ではCRC32をデータとして取り扱い、CRC32の結果まで含めてエラー検出します。

　このため、CRC16ではCRC32の演算結果が必要であるのに対して、CRC32ではCRC16の演算結果は必要ありません。簡単に言ってしまえば、データの送受信においてCRC32が先か後かの違いです。CRC16が先にあるようなフォーマットではCRC16の結果をCRC32で使用することになります。

　参考）2つのCRC符号を付加する理由：　InifiniBandを例に説明します。

others   InfiniBand
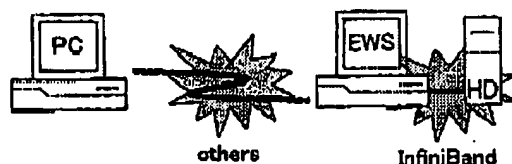
　上図のようなシステム構成で（PCとEWS間は従来の通信プロトコル（TCP/IPなど）、EWSとHD間はInfiniBandプロトコル）、PCからサーバ（EWS）を経由してHDへデータのアクセス（読み出し）を行う場合、HDからはInfiniBandプロトコルに従って前述のフォーマットでデータがEWSへ転送されます。EWSではPCへデータを送信する際に、再度CRC計算をせずにInfiniBandプロトコル専用のヘッダー情報およびCRC16を削除して送信します。この時、CRC32が付加されていなければ、CRC演算を行わなければなりません。この手間を省いています。

【発明の背景】

上述しましたようにCRC演算結果は転送データの最後に付加しなければなりません。

また、データ通信では送信の最初から最後(パケット通信では1パケット間)までは連続して転送しなければなりません。つまり、データの最後とCRC結果の間にタイムラグが発生してはいけないと言うことです。CRCの演算結果を得るためには最低でも1クロックは必要となります。このためデータ・パス組にラッチを挿入しデータの最後それに続くCRC結果が連続するようにしなければなりません。また、上述してきた理由から(片方の演算結果をもう一方で使用するため)、このCRCが2つある場合にはデータ・パス組に最低でも2クロックのラッチを挿入しなければなりません。

ここで、近年のCPUの高速化に伴って、通信分野でも単高速処理が必要不可欠となっています。高速化を実現するためには通信速度を上負ることやバス幅を広負ることが考えられます。それと同時にデバイス内部の処理速度の向上が必須となっています。

【従来の回路構成】
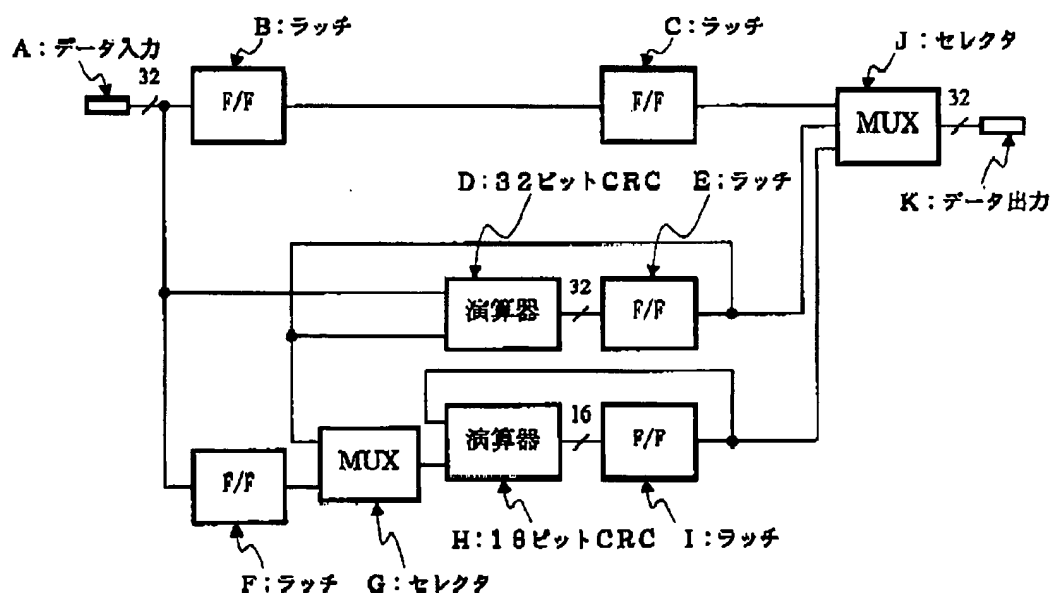
A: データ入力部
B, C: データ・パスでのタイミング調整用のラッチ(32ビット・フリップフロップ)
D: CRC32演算器
E: CRC32演算結果のラッチ(32ビット・フリップフロップ)
F: CRC16演算器へのデータ入力タイミング調整用ラッチ(t6ビット・フリップフロップ)
G: ラッチ後の入力データとCRC32の演算結果とのセレクタ回路
H: CRC16演算器
I: CRC16演算結果のラッチ(16・フリップフロップ)
J: 出力データセレクタ
K: データ出力部

　A：データ入力部からは32ビット（4バイト）データがB：ラッチ、D：CRC32演算器、およびF：ラッチ（CRC16演算器タイミング調整用）に入力されます。CRC32の演算結果はE：ラッチ回路を経て、J：出力セレクタ部へと入力される経路と、F：ラッチからの入力データとのG：セレクタ回路への入力となります。G：セレクタ回路では入力データとCRC32の演算結果から、H：CRC16演算器への入力データの選択を行います。CRC16の演算結果はI：ラッチ回路を経て、：出力セレクタ部へと入力されます。

A：データ入力

	A	B	C	D
1byte	data0	data4	data8	data12
	data1	data5	data9	data13
	data2	data6	data10	
	data3	data7	data11	

E：FF出力 — a b c d　演算結果(CRC32)

B,F：FF出力 — A B C D

G：MUX出力 — A B C D E

	A	B	C	D	E
1byte	data0	data4	data8	data12	CRC32
	data1	data5	data9	data13	CRC32
	data2	data6	data10	CRC32	
	data3	data7	data11	CRC32	

I：FF出力 — a b c d e　演算結果(CRC16)

K：データ出力 — A B C D E

	A	B	C	D	E
1byte	data0	data4	data8	data12	CRC32
	data1	data5	data9	data13	CRC32
	data2	data6	data10	CRC32	CRC16
	data3	data7	data11	CRC32	CRC16

　A：データ入力部から上図のようにデータが入力されてくるものとします。この時、入力データの最後「まるD」は2バイト（16ビット）のみとします。CRC32の演算は入力データ「まるA」とE：ラッチの初期値で最初の計算がされます。その結果をE：ラッチ部で保持し、「まるB」と前回の結果（ラッチしたデータ）とで2回目の演算を行います。これを繰り返すことで最終的にCRC32符号ビット（「まるd」）を得ることができます。

　次にCRC16への入力データはA：データ入力部からのデータの最後「まるD」に、続けてCRC32演算結果「まるd」を付加して入力しなければなりません。この時「まるD」が2バイトのみであった場合には、上図タイミングのようにCRCを2回に分けて入力しなければなりません。このためには入力データを1クロック遅らせる必要があります。このためにG：MUX部の入力側にF：ラッチを設けています。CRC16の演算もCRC32と同様に行います。これにより、最終的にCRC16符号ビット（「まるe」）を得ることができます。

　本回路構成では入力から出力までに2クロックの遅れが生じます。

【発明の回路構成】
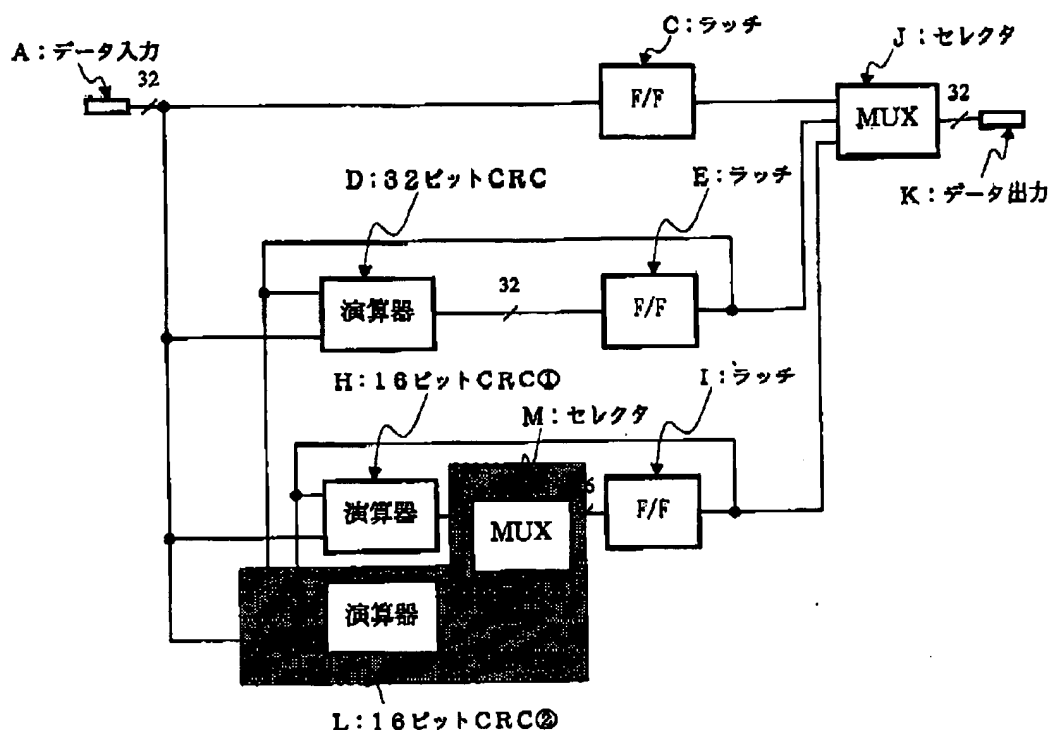
A: データ入力部
C: データ・パスでのタイミング調整用のラッチ（32ビット・フリップフロップ）
D: CRC32演算器
E: CRC32演算結果のラッチ（32ビット・フリップフロップ）
H: CRC16演算器①
I: CRC16演算結果のラッチ（16・フリップフロップ）
J: 出力データセレクタ
K: データ出力部
L: CRC16演算器②（レイテンシー削減用）
M: CRC16演算器①②からの出力セレクタ

　A：データ入力部からは32ビット（8バイト）データがC：ラッチ、D：CRC32演算器、H：CRC16演算器①およびL：CRC16演算器②に入力される。この演算器から得られた結果は各々のラッチ：E,Iを経て、出力セレクタ部：Jへと入力される。

A：データ入力

1byte	data0	data4	data8	data12
	data1	data5	data9	data13
	data2	data6	data10	
	data3	data7	data11	

演算結果(CRC32)

E：FF出力

H：CRC16①出力

L：CRC16②出力

演算結果(CRC16)

I：FF出力

J：データ出力

1byte	data0	data4	data8	data12	CRC32
	data1	data5	data9	data13	CRC32
	data2	data6	data10	CRC32	CRC16
	data3	data7	data11	CRC32	CRC16
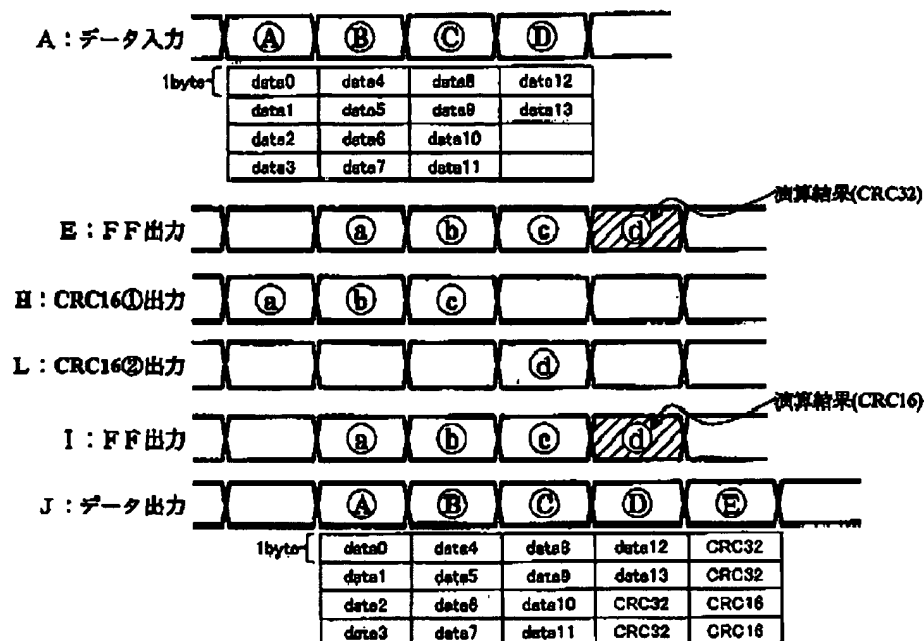
A：データ入力部から上図のようにデータが入力されてくるものとします。（従来例と同様）

CRC32の演算は入力データ「まるA」とE：ラッチの初期値で最初の計算がされます。その結果をE：ラッチ部で保持し、「まるB」と前回の結果（ラッチしたデータ）とで2回目の演算を行います。これを繰り返すことで最終的にCRC32符号ビット（「まるd」）を得ることができます。

次にCRC16の演算もCRC32と同様に、入力データ「まるA」と I：ラッチの初期値を用いてH：CRC16①で最初の計算がされます。その結果を I：ラッチ部で保持し、「まるB」と前回の結果（ラッチしたデータ）とで2回目の演算を行います。これを最後のデータの1つ前（この例では「まるC」）まで繰り返します。入力データの最後「まるD」を検知し、M：MUXを切り替えL：CRC16②から最終的なRC16符号ビット（「まるd」）を得ます。このL：CRC16②では入力データ「まるD」と I：ラッチを経由したH：CRC16①の演算結果（「まるc」）（ここまでは従来と同じ）、それに加えタイミングを早めるためにE：ラッチを経由したD：CRC32の結果（「まるc」）を入力します。CRC32は入力データ「まるD」と1つ前の演算結果「まるc」から得られます。すなわち、CRC32の演算をCRC16の計算に包括するようにしてあげれば、CRC32の結果を待たずに演算可能となります。これにより、CRC32の演算結果を待たずにすみますからCRC16の演算結果は従来に比べ1クロック（MIN.）前に得ることができます。よって、データ・バス側も1クロックの遅れのみとなります。

本回路構成では入力から出力までに1クロックの遅れが生じます。これは従来までの演算式を用いた回路構成に比べ1クロック（MIN.）分のレイテンシー削減となります。

【演算式算出方法】

従来例および発明例で使用するCRC演算器は以下の演算式を使用します。

D:CRC32演算器 ・・・P6～P8に記載した演算式を使用します。

H:CRC16演算器 ・・・P5に記載した生成回路を利用して、32シフトした場合の各FF出力を使用します。なお、P5に記載しました演算式は16シフト時のものです。

L:CRC16演算器 ・・・新たに以下の方法で演算式を算出。

上述の【発明の回路構成】の項でも述べたように、CRC32の演算結果を得てからCRC16の計算を行ったのでは、どうしてもそこに1クロック分の遅れが生じてしまいます。これを解消するためにCRC16の演算を行う際に一緒にCRC32の演算も行えば、遅れを生じることなく結果を得ることができます。

そこで下記手順で新たな演算式を算出し使用することで上記の遅れを解消しています。

①データ幅64ビット

CRC16演算器では入力データ(32ビット)と1つ前の演算結果（16ビット）を用いて算出します。従来方法ではこの入力データの最後（次）にCRC32符号ビット（32ビット）を付加してCRC16の符号ビットを得ています。本発明で付加したL:CRC16②では入力データと同時にCRC32の結果（最終結果の1つ前）を入力します。つまり入力データを64ビットとして演算式を算出します。この時、本来の入力データ「まるD」を下位ビット、CRC演算結果を上位ビットで入力します。

63　　　　　　　　　　　　　3231　　　　　　　　　　　　　0
CRC32演算結果　　　　　　　　入力データ

32ビット　　　　　　　　　　　　　32ビット
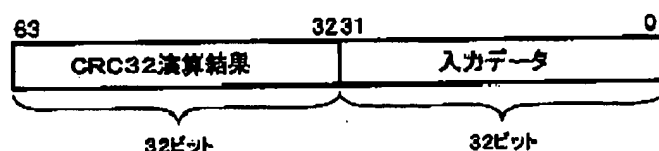
②演算式算出－Ⅰ

まず、データ幅64ビットでのCRC16演算式を算出します。これはP5記載のCRC16生成回路から、64シフトした際の各FF出力となります。

C0	R2・R4・R5・R8・R9・R11・R12・R13・R14・D1・D2・D3・D4・D6・D7・D10・D11・D13・D17・D18・D19・D20・D22・D26・D29・D31・D32・D34・D35・D37・D40・D42・D43・D48・D50・D51・D55・D59・D63
C1	R2・R3・R4・R6・R8・R10・R11・R15・D0・D4・D5・D7・D9・D11・D12・D13・D16・D20・D21・D22・D25・D26・D28・D29・D30・D32・D33・D35・D36・D37・D39・D40・D41・D43・D47・D48・D49・D51・D54・D55・D58・D59・D62・D63
C2	R0・R3・R4・R5・R7・R9・R11・R12・D3・D4・D6・D8・D10・D11・D12・D15・D19・D20・D21・D24・D25・D27・D28・D29・D31・D32・D34・D35・D36・D38・D39・D40・D42・D46・D47・D48・D50・D53・D54・D57・D58・D61・D62
C3	R1・R2・R6・R9・R10・R11・R14・D1・D4・D5・D6・D9・D13・D14・D17・D22・D23・D24・D27・D28・D29・D30・D32・D33・D38・D39・D40・D41・D42・D43・D45・D46・D47・D48・D49・D50・D51・D52・D53・D55・D56・D57・D59・D60・D61・D63

C4	R2・R3・R7・R10・R11・R12・R15・D0・D3・D4・D5・D8・D12・D13・D16・D21・D22・D23・D26・D27・D28・D29・D31・D32・D37・D38・D39・D40・D41・D42・D44・D45・D46・D47・D48・D49・D50・D51・D52・D54・D55・D56・D58・D59・D60・D62
C5	R0・R3・R4・R8・R11・R12・R13・D02・D03・D4・D7・D11・D12・D15・D20・D21・D22・D25・D26・D27・D28・D30・D31・D36・D37・D38・D39・D40・D41・D43・D44・D45・D46・D47・D48・D49・D50・D51・D53・D54・D55・D57・D58・D59・D61
C6	R1・R4・R5・R9・R12・R13・R14・D1・D2・D3・D6・D10・D11・D14・D19・D20・D21・D24・D25・D26・D27・D29・D30・D35・D36・D37・D38・D39・D40・D42・D43・D44・D45・D46・D47・D48・D49・D50・D52・D53・D54・D56・D57・D58・D60
C7	R2・R5・R6・R10・R13・R14・R15・D0・D1・D2・D5・D09・D10・D13・D18・D19・D20・D23・D24・D25・D26・D28・D29・D34・D35・D36・D37・D38・D39・D41・D42・D43・D44・D45・D46・D47・D48・D49・D51・D52・D53・D55・D56・D57・D59
C8	R3・R6・R7・R11・R14・R15・D0・D1・D4・08・D9・D12・D17・D18・D19・D22・D23・D24・D25・D27・D28・D33・D34・D35・D36・D37・D38・D40・D41・D42・D43・D44・D45・D46・D47・D48・D50・D51・D52・D54・D55・D56・D58
C9	R4・R7・R8・R12・R15・D0・D3・D7・D8・D11・D16・D17・D18・D21・D22・D23・D24・D26・D27・D32・D33・D34・D35・D36・D37・D39・D40・D41・D42・D43・D44・D45・D46・D47・D49・D50・D51・D53・D54・D55・D57
C10	R0・R5・R08・R9・R13・D2・D6・D7・D10・D15・D16・D17・D20・D21・D22・D23・D25・D26・D31・D32・D33・D34・D35・D36・D38・D39・D40・D41・D42・D43・D44・D45・D46・D48・D49・D50・D52・D53・D54・D56
C11	R0・R1・R6・R9・R10・R14・D1・D5・D6・D9・D14・D15・D16・D19・D20・D21・D22・D24・D25・D30・D31・D32・D33・D34・D35・D37・D38・D39・D40・D41・D42・D43・D44・D45・D47・D48・D49・D51・D52・D53・D55
C12	R0・R1・R4・R5・R7・R8・R9・R10・R12・R13・R14・R15・D0・D1・D2・D3・D5・D6・D7・D8・D10・D11・D14・D15・D17・D21・D22・D23・D24・D26・D30・D33・D35・D36・D38・D39・D41・D44・D46・D47・D52・D54・D55・D59・D63
C13	R1・R2・R5・R6・R8・R9・R10・R11・R13・R14・R15・D0・D1・D2・D4・D5・D6・D7・D9・D10・D13・D14・D16・D20・D21・D22・D23・D25・D29・D32・D34・D35・D37・D38・D40・D43・D45・D46・D51・D53・D54・D58・D62
C14	R0・R2・R3・R6・R7・R9・R10・R11・R12・R14・R15・D0・D1・D3・D4・D5・D6・D8・D9・D12・D13・D15・D19・D20・D21・D22・D24・D28・D31・D33・D34・D36・D37・D39・D42・D44・D45・D50・D52・D53・D57・D61
C15	R1・R3・R4・R7・R8・R10・R11・R12・R13・R15・D0・D2・D3・D4・D5・D7・D8・D11・D12・D14・D18・D19・D20・D21・D23・D27・D30・D32・D33・D35・D36・D38・D41・D43・D44・D49・D51・D52・D56・D60

③データ置換

　②で得られた演算式に、①よりD63～D31まではCRC32の演算結果ですのでP7，8に記載した演算式を代入します。

例)最下位ビット(C0)での置き換え例

C0= R2・R4・R5・R8・R9・R11・R12・R13・R14・
D1・D2・D3・D4・D6・D7・D10・D11・D13・D17・D18・D19・D20・D22・D26・D29・D31・
D32・D34・D35・D37・D40・D42・D43・D48・D50・D51・D55・D59・D63

P7, 8より

D63 <=   C31=   R5・R8・R9・R11・R15・R23・R24・R25・R27・R28・R29・R30・R31・
D0・D1・D2・D3・D4・D6・D7・D8・D16・D20・D22・D23・D26

D59 <=   C27=   R1・R4・R5・R7・R11・R19・R20・R21・R23・R24・R25・R26・R27・
R29・
D2・D4・D5・D6・D7・D8・D10・D11・D12・D20・D24・D26・D27・D30

D55 <=   C23=   R0・R1・R6・R9・R13・R15・R16・R17・R19・R20・R26・R27・R29・
R31・
D0・D2・D4・D5・D11・D12・D14・D15・D16・D18・D22・D25・D30・D31

D51 <=   C19=   R3・R7・R8・R11・R15・R16・R20・R22・R24・R25・R27・R29・
D2・D4・D6・D7・D9・D11・D15・D16・D20・D23・D24・D28

D50 <=   C18=   R2・R6・R7・R10・R14・R15・R19・R21・R23・R24・R26・R28・R31・
D0・D3・D5・D7・D8・D10・D12・D16・D17・D21・D24・D25・D29

D48 <=   C16=   R0・R4・R5・R8・R12・R13・R17・R19・R21・R22・R24・R26・R29・
R30・
D1・D2・D5・D7・D9・D10・D12・D14・D18・D19・D23・D26・D27・D31

D43 <=   C11=   R0・R1・R3・R4・R9・R12・R14・R15・R16・R17・R20・R24・R25・
R26・R27・R28・R31・
D0・D3・D4・D5・D6・D7・D11・D14・D15・D16・D17・D19・D22・D27・D28・D30・
D31

D42 <=   C10=   R0・R2・R3・R5・R9・R13・R14・R16・R19・R26・R28・R29・R31・
D0・D2・D3・D5・D12・D15・D17・D18・D22・D26・D28・D29・D31

D40 <=   C8=R0・R1・R3・R4・R8・R10・R11・R17・R22・R28・R31・
D0・D3・D8・D9・D14・D19・D20・D21・D23・D27・D28・D30・D31

D37 <=   C5=R0・R1・R3・R4・R5・R6・R7・R10・R13・R19・R20・R21・R24・R28・R29・
D2・D3・D7・D10・D11・D12・D18・D21・D24・D25・D26・D27・D28・D30・D31

D35 <=   C3=R1・R2・R3・R7・R8・R9・R10・R14・R15・R17・R18・R19・R25・R27・R31・
D0・D4・D6・D12・D13・D14・D16・D17・D21・D22・D23・D24・D28・D29・D30

D34 <=   C2=R0・R2・R6・R7・R8・R9・R13・R14・R16・R17・R18・R24・R26・R30・R31・
D0・D1・D5・D7・D13・D14・D15・D17・D18・D22・D23・D24・D25・D29・D30・
D31

D32 <=   C0=R0・R6・R9・R10・R12・R16・R24・R25・R26・R28・R29・R30・R31・
D0・D1・D2・D3・D5・D6・D7・D15・D19・D21・D22・D25・D31

代入して

$C_0 =$ R2・R4・R5・R8・R9・R11・R12・R13・R14・
D1・D2・D3・D4・D6・D7・D10・D11・D13・D17・D18・D19・D20・D22・D26・D29・D31・
R5・R8・R9・R11・R15・R23・R24・R25・R27・R28・R29・R30・R31・
D0・D1・D2・D3・D4・D6・D7・D8・D16・D20・D22・D23・D26
R1・R4・R5・R7・R11・R19・R20・R21・R23・R24・R25・R26・R27・R29・
D2・D4・D5・D6・D7・D8・D10・D11・D12・D20・D24・D26・D27・D30
R0・R1・R6・R9・R13・R15・R16・R17・R19・R20・R26・R27・R29・R31・
D0・D2・D4・D5・D11・D12・D14・D15・D16・D18・D22・D25・D30・D31
R3・R7・R8・R11・R15・R16・R20・R22・R24・R25・R27・R29・
D2・D4・D6・D7・D9・D11・D15・D16・D20・D23・D24・D28
R2・R6・R7・R10・R14・R15・R19・R21・R23・R24・R26・R28・R31・
D0・D3・D5・D7・D8・D10・D12・D16・D17・D21・D24・D25・D29
R0・R4・R5・R8・R12・R13・R17・R19・R21・R22・R24・R26・R29・R30・
D1・D2・D5・D7・D9・D10・D12・D14・D18・D19・D23・D26・D27・D31
R0・R1・R3・R4・R9・R12・R14・R15・R16・R17・R20・R24・R25・R26・R27・R28・R31・
D0・D3・D4・D5・D6・D7・D11・D14・D15・D16・D17・D19・D22・D27・D28・D30・D31
R0・R2・R3・R5・R9・R13・R14・R16・R19・R26・R28・R29・R31・
D0・D2・D3・D5・D12・D15・D17・D18・D22・D26・D28・D29・D31
R0・R1・R3・R4・R8・R10・R11・R17・R22・R28・R31・
D0・D3・D8・D9・D14・D19・D20・D21・D23・D27・D28・D30・D31
R0・R1・R3・R4・R5・R6・R7・R10・R13・R19・R20・R21・R24・R28・R29・
D2・D3・D7・D10・D11・D12・D18・D21・D24・D25・D26・D27・D28・D30・D31
R1・R2・R3・R7・R8・R9・R10・R14・R15・R17・R18・R19・R25・R27・R31・
D0・D4・D6・D12・D13・D14・D16・D17・D21・D22・D23・D24・D28・D29・D30
R0・R2・R6・R7・R8・R9・R13・R14・R16・R17・R18・R24・R26・R30・R31・
D0・D1・D5・D7・D13・D14・D15・D17・D18・D22・D23・D24・D25・D29・D30・D31
R0・R6・R9・R10・R12・R16・R24・R25・R26・R28・R29・R30・R31・
D0・D1・D2・D3・D5・D6・D7・D15・D19・D21・D22・D25・D31

同一項を削除して（青のRxxはCRC16の演算結果：赤のRxxはCRC32の演算結果ですので、同
一項でも削除不可、わかりやすくするためにCRC16の演算結果（青）はZxxに置き換えます。）

$C_0 =$ Z2・Z4・Z5・Z8・Z9・Z11・Z12・Z13・Z14・
R1・R4・R5・R6・R9・R10・R13・R14・R19・R20・R22・R24・R28・R31・
D0・D1・D2・D4・D6・D9・D10・D12・D13・D19・D20・D21・D25・D27・D29・D30・D31・

以上を全ての演算式において行うことで、CRC32とCRC16の最終的な演算結果を同時に得ることが
可能となります。

　　　実施例2として、CRC演算器を3つ必要とする場合のデータ・フォーマットおよび回路構成、タイミング、演算式に関して以降に示します。

【データ・フォーマット】

　　　CRC16を3つ必要とする場合に関して説明します。実施例2ではCRC16を3つ必要とするような場合に関してのみ説明いたしますが、他の演算（CRC32など）を用いた場合でも新たに算出する演算式を以降に示すアルゴリズムに則って生成することで、同様の回路構成で対応が可能となります。
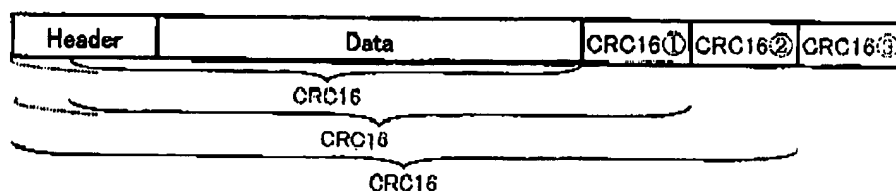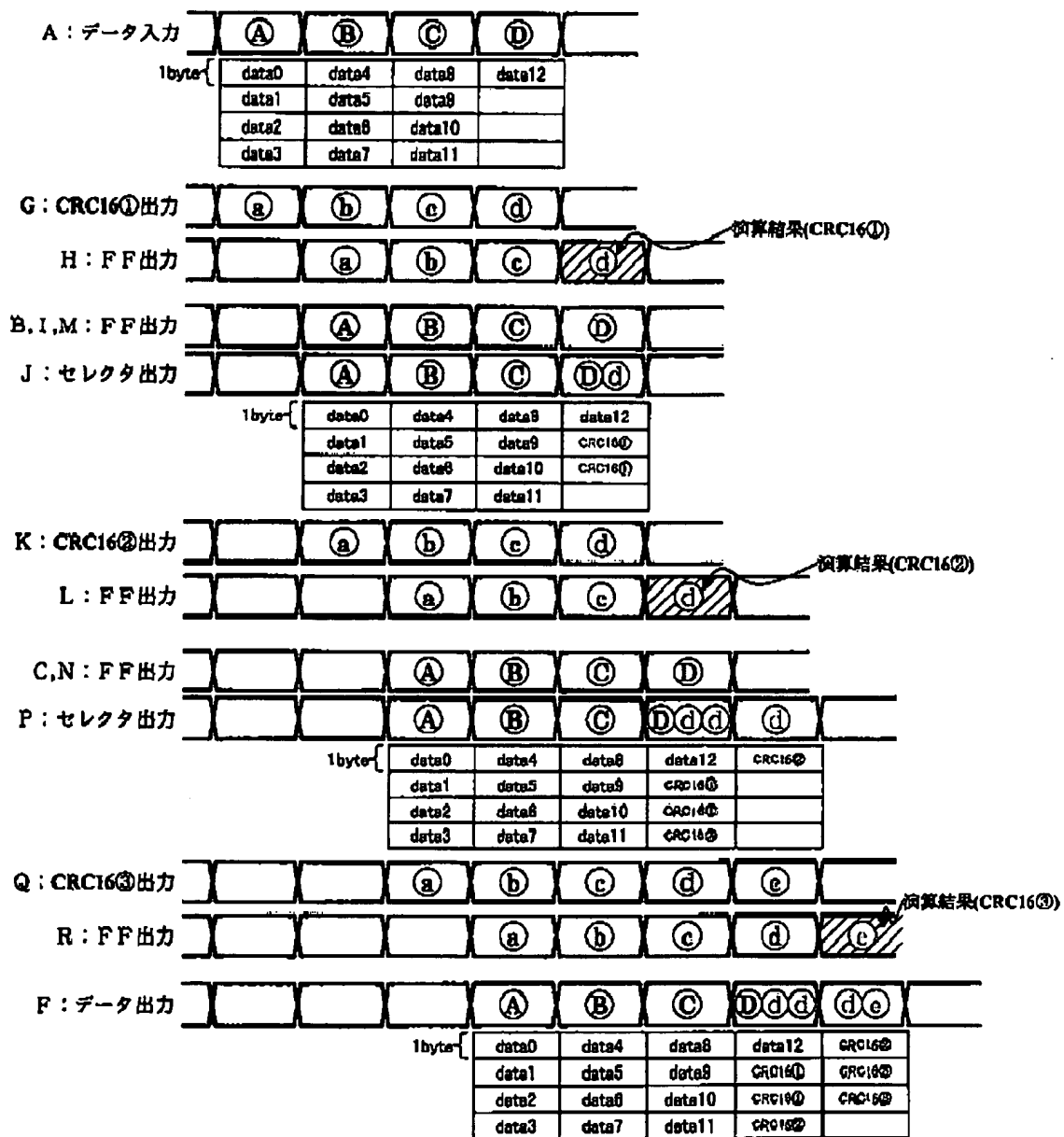
Header	Data	CRC16①	CRC16②	CRC16③

CRC16
CRC16
CRC16

例) 4バイト（32ビット）転送

1byte

#1	data0	data1	data2	data3
#2	data4	data5	data6	data7
#3	data8	data9	data10	data11

⋮

#n-2	data n-4	data n-3	data n-2	data n-1
#n-1	data n	CRC16①	CRC16①	CRC16②
#n	CRC16②	CRC16③	CRC16③	

　　　CRCは転送データのエラーを検出するためのものです。つまり、上記のようなデータを転送する場合、data0～data nまでのエラー検出に使用されます。CRC16②ではCRC16①をデータとして取り扱い、CRC16①の結果まで含めてエラー検出します。また、CRC16③ではCRC16①、CRC16②をデータとして取り扱い、CRC16②の結果まで含めてエラー検出します。
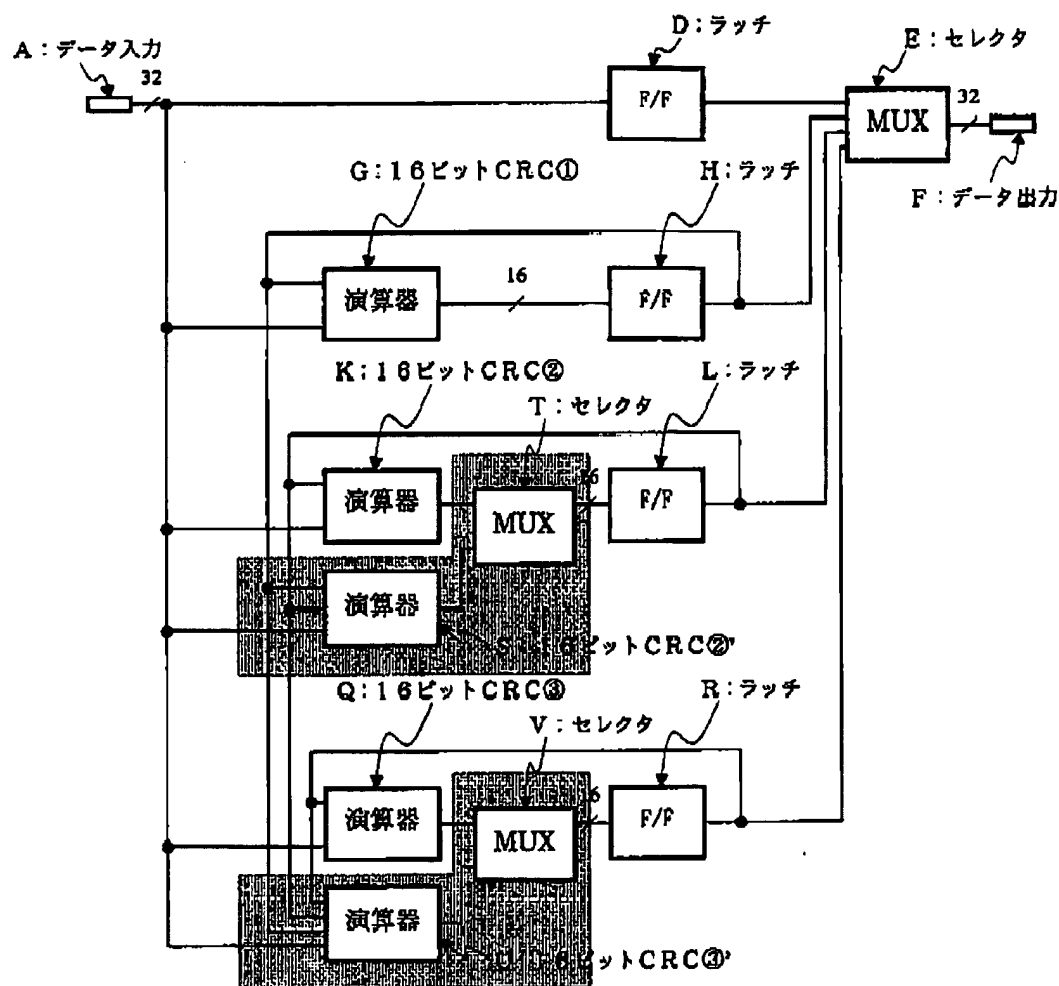
【従来方法による回路構成】
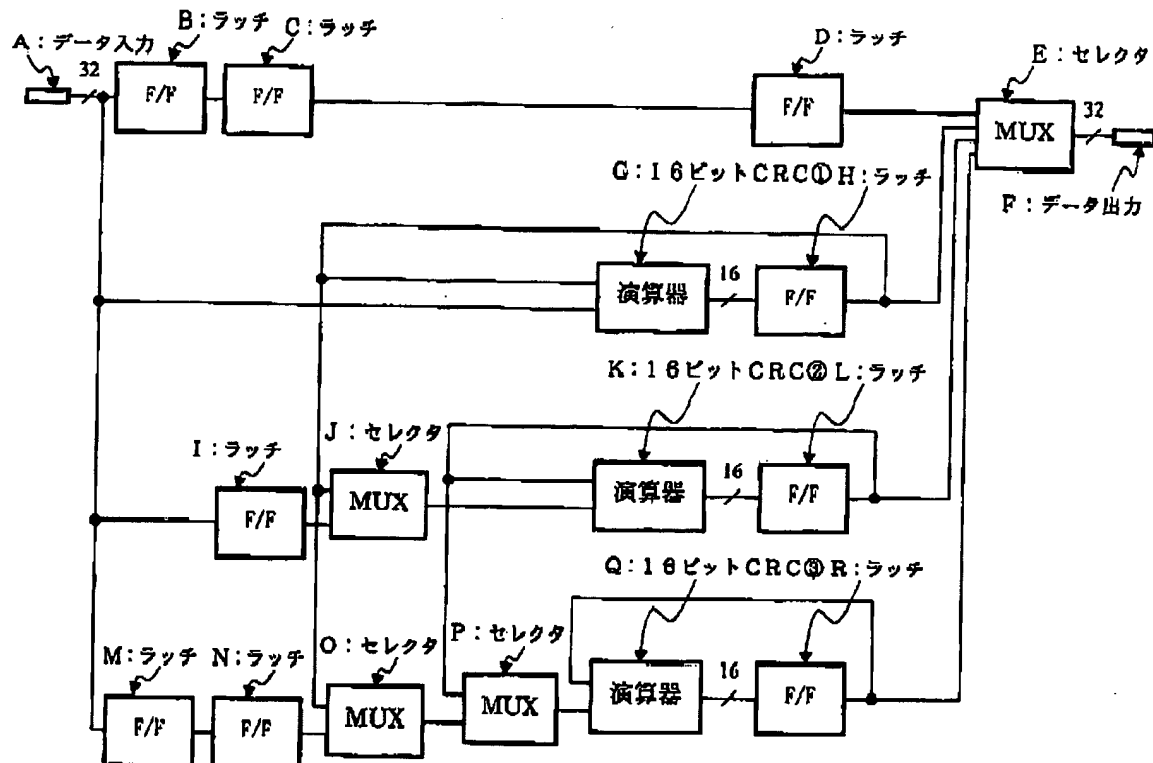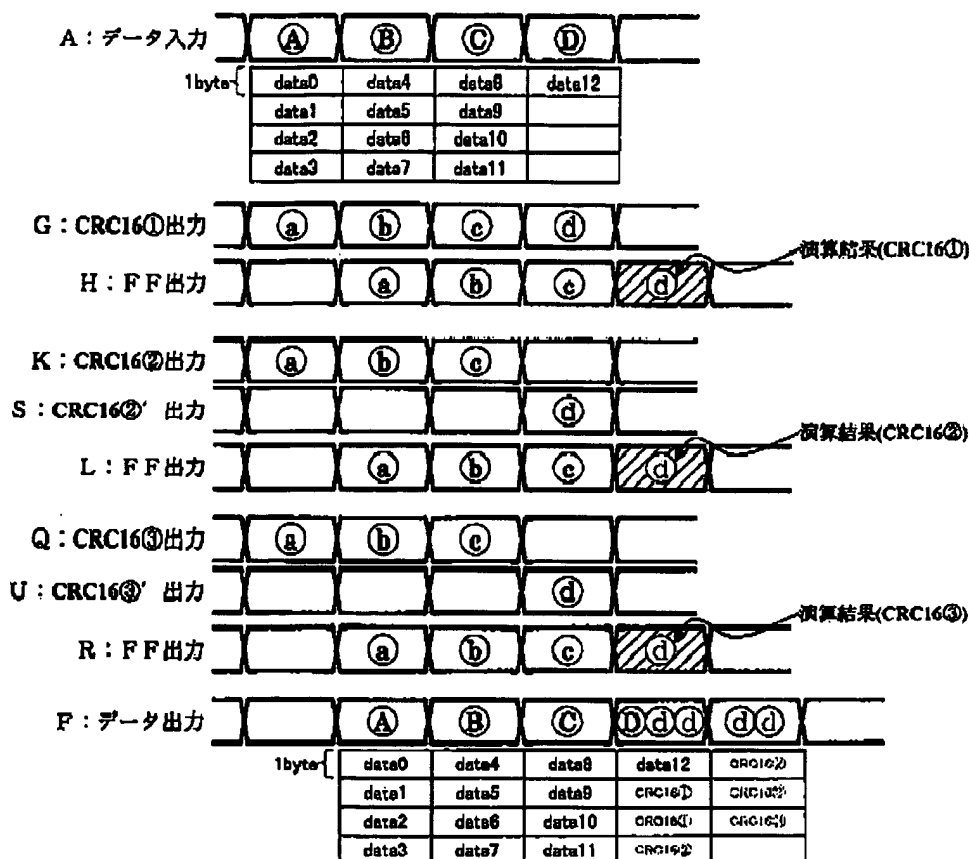

A:　データ入力部
B,C,D:　データ・バスでのタイミング調整用のラッチ（32ビット・フリップフロップ）
E:　出力データセレクタ
F:　データ出力部
G:　CRC16演算器①
H:　CRC16①演算結果のラッチ（16ビット・フリップフロップ）
I:　CRC16②へのタイミング調整用のラッチ（32ビット・フリップフロップ）
J,O:　入力データ、CRC16①演算結果のセレクタ
K:　CRC16演算器②
L:　CRC16②演算結果のラッチ（16・フリップフロップ）
L:　CRC16演算器②（レイテンシー削減用）
M,N:　CRC16②へのタイミング調整用のラッチ（32ビット・フリップフロップ）
P:　セレクタ：O出力とCRC16②演算結果のセレクタ
Q:　CRC16演算器③
R:　CRC16③演算結果のラッチ（16・フリップフロップ）

A：データ入力　Ⓐ Ⓑ Ⓒ Ⓓ

1byte			
data0	data4	data8	data12
data1	data5	data9	
data2	data6	data10	
data3	data7	data11	

G：CRC16① 出力　ⓐ ⓑ ⓒ ⓓ

H：FF出力　ⓐ ⓑ ⓒ ⓓ　演算結果(CRC16①)

B,I,M：FF出力　Ⓐ Ⓑ Ⓒ Ⓓ

J：セレクタ出力　Ⓐ Ⓑ Ⓒ Ⓓⓓ

1byte			
data0	data4	data8	data12
data1	data5	data9	CRC16①
data2	data6	data10	CRC16①
data3	data7	data11	

K：CRC16② 出力　ⓐ ⓑ ⓒ ⓓ

L：FF出力　ⓐ ⓑ ⓒ ⓓ　演算結果(CRC16②)

C,N：FF出力　Ⓐ Ⓑ Ⓒ Ⓓ

P：セレクタ出力　Ⓐ Ⓑ Ⓒ Ⓓⓓⓓ ⓓ

1byte				
data0	data4	data8	data12	CRC16②
data1	data5	data9	CRC16①	
data2	data6	data10	CRC16①	
data3	data7	data11	CRC16②	

Q：CRC16③ 出力　ⓐ ⓑ ⓒ ⓓ ⓔ

R：FF出力　ⓐ ⓑ ⓒ ⓓ ⓔ　演算結果(CRC16③)

F：データ出力　Ⓐ Ⓑ Ⓒ Ⓓⓓⓓ ⓓⓔ

1byte				
data0	data4	data8	data12	CRC16②
data1	data5	data8	CRC16①	CRC16②
data2	data6	data10	CRC16①	CRC16②
data3	data7	data11	CRC16②	

【実施例2の回路構成】

A: データ入力部
D: データ・パスでのタイミング調整用のラッチ（32ビット・フリップフロップ）
E: 出力データセレクタ
F: データ出力部
G: CRC16演算器①
H: CRC16①演算結果のラッチ（16ビット・フリップフロップ）
K: CRC16演算器②
L: CRC16②演算結果のラッチ（16ビット・フリップフロップ）
Q: CRC16演算器③
R: CRC16③演算結果のラッチ（16ビット・フリップフロップ）
S: CRC16演算器②'（レイテンシー削減用）
T: CRC16演算器②②'からの出力セレクタ
U: CRC16演算器③'（レイテンシー削減用）
V: CRC16演算器③③'からの出力セレクタ

A：データ入力

1byte	Ⓐ	Ⓑ	Ⓒ	Ⓓ
	data0	data4	data8	data12
	data1	data5	data9	
	data2	data6	data10	
	data3	data7	data11	

G：CRC16①出力　　ⓐ　ⓑ　ⓒ　ⓓ

H：FF出力　　　　ⓐ　ⓑ　ⓒ　ⓓ（斜線）　→ 演算結果(CRC16①)

K：CRC16②出力　　ⓐ　ⓑ　ⓒ

S：CRC16②'出力　　　　　　ⓓ　→ 演算結果(CRC16②)

L：FF出力　　　　ⓐ　ⓑ　ⓒ　ⓓ（斜線）

Q：CRC16③出力　　ⓐ　ⓑ　ⓒ

U：CRC16③'出力　　　　　　ⓓ　→ 演算結果(CRC16③)

R：FF出力　　　　ⓐ　ⓑ　ⓒ　ⓓ（斜線）

F：データ出力　　Ⓐ　Ⓑ　Ⓒ　Ⓓⓓⓓ　ⓓⓓ

1byte	data0	data4	data8	data12	CRC16②
	data1	data5	data9	CRC16①	CRC16③
	data2	data6	data10	CRC16①	CRC16③
	data3	data7	data11	CRC16②	

A：データ入力部から上図のようにデータが入力されてくるものとします。（従来例と同様）

　CRC16①の演算は入力データ「まるA」とH：ラッチの初期値で最初の計算がされます。その結果をH：ラッチ部で保持し、「まるB」と前回の結果（ラッチしたデータ）とで2回目の演算を行います。これを繰り返すことで最終的にCRC16①符号ビット（「まるd」）を得ることができます。

　次にCRC16②の演算も①と同様に、入力データ「まるA」と　L：ラッチの初期値を用いてK：CRC16②で最初の計算がされます。その結果を　L：ラッチ部で保持し、「まるB」と前回の結果（ラッチしたデータ）とで2回目の演算を行います。これを最後のデータの1つ前（この例では「まるC」）まで繰り返します。入力データの最後「まるD」を検知し、T：MUXを切り替えS：CRC16②'　から最終的なRC16②符号ビット（「まるd」）を得ます。このS：CRC16②'　では入力データ「まるD」と　L：ラッチを経由したK：CRC16②の演算結果（「まるc」）（ここまでは従来と同じ）、それに加えタイミングを早めるためにH：ラッチを経由したG：CRC16①の結果（「まるc」）を入力します。CRC16①は入力データ「まるD」と1つ前の演算結果「まるc」から得られます。すなわち、CRC16①演算をCRC16②の計算に包括するようにしてあげれば、CRC16①の結果を待たずに演算可能となります。これにより、CRC16①の演算結果を待たずにすみますからCRC16②の演算結果は従来に比べ1クロック（MIN.）前に得ることができます。よって、データ・パス側も1クロックの遅れのみとなります。

　さらに、3番目のCRC16演算ではCRC16③'　において、上記と同様の方法でCRC16①および②の演算を包括するようにしてあげれば、CRC16①および②の結果を待たずに演算可能となります。

　本回路構成では入力から出力までに1クロックの遅れが生じます。これは従来までの演算式を用いた回路構成に比べ2クロック（MIN.）分のレイテンシー削減となります。

－5－

【演算式算出方法】

従来例および発明例で使用するCRC演算器は以下の演算式を使用します。

G, K, Q: CRC16演算器 ・・・ 先般お渡ししました資料P5に記載した生成回路を利用して、32シフトした場合の各FF出力を使用します。本資料のP7に記載。

S: CRC16演算器 ・・・ 先般お渡ししました資料の演算式算出方法で、本資料P11に記載の演算式を使用します。

U: CRC16演算器 ・・・ 新たに以下の方法で演算式を算出。

CRC16①および②の演算結果を得てからCRC16③の計算を行ったのでは、どうしてもそこに2クロック分の遅れが生じてしまいます。これを解消するためにCRC16②の演算を行う際に一緒にCRC16①の演算も行います（先般の資料で説明）。さらに、CRC16③の演算を行う際にCRC16①および②の演算を行えば、各々のCRC演算結果を待たずに、各々同時に結果を得ることが可能となります。

そこで下記手順で新たな演算式を算出し使用することで上記の遅れを解消しています。

①G：CRC16演算器、K：CRC16演算器②、Q：CRC16演算器③　演算式算出

先般お渡ししました資料P5に記載した生成回路を利用して、32シフトした場合の各FF出力

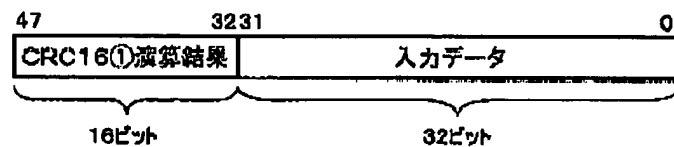|	演算式　（・：ExculsiveOR）
C15	X03・X04・X06・X09・X11・X12・X14・X15・ D00・D01・D03・D04・D06・D09・D11・D12・D17・D19・D20・D24・D28
C14	X02・X03・X05・X08・X10・X11・X13・X14・ D01・D02・D04・D05・D07・D10・D12・D13・D18・D20・D21・D25・D29
C13	X01・X02・X04・X07・X09・X10・X12・X13・X15・ D00・D02・D03・D05・D06・D08・D11・D13・D14・D19・D21・D22・D26・D30
C12	X00・X01・X03・X06・X08・X09・X11・X12・X14・ D01・D03・D04・D06・D07・D09・D12・D14・D15・D20・D22・D23・D27・D31
C11	X00・X02・X03・X04・X05・X06・X07・X08・X09・X10・X12・X13・X14・X15・ D00・D01・D02・D03・D05・D06・D07・D08・D09・D10・D11・D12・D13・D15・D16・D17・D19・D20・D21・ D23
C10	X01・X02・X03・X04・X05・X06・X07・X08・X09・X11・X12・X13・X14・X15・ D00・D01・D02・D03・D04・D06・D07・D08・D09・D10・D11・D12・D13・D14・D16・D17・D18・D20・D21・ D22・D24
C09	X00・X01・X02・X03・X04・X05・X06・X07・X08・X10・X11・X12・X13・X14・X15・ D00・D01・D02・D03・D04・D05・D07・D08・D09・D10・D11・D12・D13・D14・D15・D17・D18・D19・D21・ D22・D23・D25
C08	X00・X01・X02・X03・X04・X05・X06・X07・X09・X10・X11・X12・X13・X14・ D01・D02・D03・D04・D05・D06・D08・D09・D10・D11・D12・D13・D14・D15・D16・D18・D19・D20・D22・ D23・D24・D26
C07	X00・X01・X02・X03・X04・X05・X06・X08・X09・X10・X11・X12・X13・ D02・D03・D04・D05・D06・D07・D09・D10・D11・D12・D13・D14・D15・D16・D17・D19・D20・D21・D23・ D24・D25・D27
C06	X00・X01・X02・X03・X04・X05・X07・X08・X09・X10・X11・X12・ D03・D04・D05・D06・D07・D08・D10・D11・D12・D13・D14・D15・D16・D17・D18・D20・D21・D22・D24・ D25・D26・D28
C05	X00・X01・X02・X03・X04・X06・X07・X08・X09・X10・X11・ D04・D05・D06・D07・D08・D09・D11・D12・D13・D14・D15・D16・D17・D18・D19・D21・D22・D23・D25・ D26・D27・D29
C04	X00・X01・X02・X03・X05・X06・X07・X08・X09・X10・X15・ D00・D05・D06・D07・D08・D09・D10・D12・D13・D14・D15・D16・D17・D18・D19・D20・D22・D23・D24・ D26・D27・D28・D30
C03	X00・X01・X02・X04・X05・X06・X07・X08・X09・X14・X15・ D00・D01・D06・D07・D08・D09・D10・D11・D13・D14・D15・D16・D17・D18・D19・D20・D21・D23・D24・ D25・D27・D28・D29・D31
C02	X00・X01・X05・X07・X08・X09・X11・X12・X13・X15・ D00・D02・D03・D04・D06・D07・D08・D10・D14・D15・D16・D18・D21・D22・D25・D26・D29・D30
C01	X00・X04・X06・X07・X08・X10・X11・X12・X14・X15・ D00・D01・D03・D04・D05・D07・D08・D09・D11・D15・D16・D17・D19・D22・D23・D26・D27・D30・D31
C00	X04・X05・X07・X10・X12・X13・X15・ D00・D02・D03・D05・D08・D10・D11・D16・D18・D19・D23・D27・D31

なお、XはH：ラッチ（フリップ・フロップ）の初期値を意味します。

②S：CRC16演算器②′ 演算式算出

先般お渡しした資料と同様の方法で、演算式の算出を行います。

a. データ幅48ビット

CRC16演算器②′では入力データ（32ビット）と1つ前の演算結果（16ビット）を用いて算出します。従来方法ではこの入力データの最後（次）にCRC16符号ビット（CRC16演算器①の結果：16ビット）を付加してCRC16の符号ビットを得ています。本発明で付加したS：CRC16演算器②′では入力データと同時にCRC16演算器①の結果（最終結果の1つ前）を入力します。つまり入力データを48ビットとして演算式を算出します。この時、本来の入力データを下位ビット、CRC演算結果を上位ビットで入力します。

```
 47           3231                          0
┌────────────┬──────────────────────────┐
│CRC16①演算結果│         入力データ          │
└────────────┴──────────────────────────┘
     16ビット                 32ビット
```

b. 演算式算出－I

まず、データ幅48ビットでのCRC16演算式を算出します。これは先般お渡しした資料P5記載のCRC16生成回路から、48シフトした際の各FF出力となります。

	演算式　（・：ExcursiveOR）
C15	Z01・Z04・Z08・Z10・Z11・Z12・Z13・D02・D03・D04・D05・D07・D11・D14・D16・D17・D19・D20・D22・D25・D27・D28・D33・D35・D36・D40・D44
C14	Z00・Z03・Z07・Z09・Z10・Z11・Z12・D03・D04・D05・D06・D08・D12・D15・D17・D18・D20・D21・D23・D26・D28・D29・D34・D36・D37・D41・D45
C13	Z02・Z06・Z08・Z09・Z10・Z11・Z15・D00・D04・D05・D06・D07・D09・D13・D16・D18・D19・D21・D22・D24・D27・D29・D30・D35・D37・D38・D42・D46
C12	Z01・Z05・Z07・Z08・Z09・Z10・Z14・D01・D05・D06・D07・D08・D10・D14・D17・D19・D20・D22・D23・D25・D28・D30・D31・D36・D38・D39・D43・D47
C11	Z00・Z01・Z06・Z07・Z09・Z10・Z11・Z12・Z15・D00・D03・D04・D05・D06・D08・D09・D14・D15・D16・D17・D18・D19・D21・D22・D23・D24・D25・D26・D27・D28・D29・D31・D32・D33・D35・D36・D37・D39
C10	Z00・Z05・Z06・Z08・Z09・Z10・Z11・Z14・Z15・D00・D01・D04・D05・D06・D07・D09・D10・D15・D16・D17・D18・D19・D20・D22・D23・D24・D25・D26・D27・D28・D29・D30・D32・D33・D34・D37・D36・D38・D40
C09	Z04・Z05・Z07・Z08・Z09・Z10・Z13・Z14・Z15・D00・D01・D02・D05・D06・D07・D08・D10・D11・D16・D17・D18・D19・D20・D21・D23・D24・D25・D26・D27・D28・D29・D30・D31・D33・D34・D35・D37・D38・D39・D41
C08	Z03・Z04・Z06・Z07・Z08・Z09・Z12・Z13・Z14・D01・D02・D03・D06・D07・D08・D09・D11・D12・D17・D18・D19・D20・D21・D22・D24・D25・D26・D27・D28・D29・D30・D31・D32・D34・D35・D36・D38・D39・D40・D42
C07	Z02・Z03・Z05・Z06・Z07・Z08・Z11・Z12・Z13・D02・D03・D04・D07・D08・D09・D10・D12・D13・D18・D19・D20・D21・D22・D23・D25・D26・D27・D28・D29・D30・D31・D32・D33・D35・D36・D37・D39・D40・D41・D43
C06	Z01・Z02・Z04・Z05・Z06・Z07・Z10・Z11・Z12・D03・D04・D05・D08・D09・D10・D11・D13・D14・D19・D20・D21・D22・D23・D24・D26・D27・D28・D29・D30・D31・D32・D33・D34・D36・D37・D38・D40・D41・D42・D44
C05	Z00・Z01・Z03・Z04・Z05・Z06・Z09・Z10・Z11・D04・D05・D06・D09・D10・D11・D12・D14・D15・D20・D21・D22・D23・D24・D25・D27・D28・D29・D30・D31・D32・D33・D34・D35・D37・D38・D39・D41・D42・D43・D45
C04	Z00・Z02・Z03・Z04・Z05・Z08・Z09・Z10・Z15・D00・D05・D06・D07・D10・D11・D12・D13・D15・D16・D21・D22・D23・D24・D25・D26・D28・D29・D30・D31・D32・D33・D34・D35・D36・D38・D39・D40・D42・D43・D44・D46
C03	Z01・Z02・Z03・Z04・Z07・Z08・Z09・Z14・D01・D06・D07・D08・D11・D12・D13・D14・D16・D17・D22・D23・D24・D25・D26・D27・D29・D30・D31・D32・D33・D34・D35・D36・D37・D39・D40・D41・D43・D44・D45・D47
C02	Z00・Z02・Z03・Z04・Z06・Z07・Z10・Z11・Z12・D03・D04・D05・D08・D09・D11・D12・D13・D15・D16・D18・D19・D20・D22・D23・D24・D26・D30・D31・D32・D34・D37・D38・D41・D42・D45・D46・
C01	Z01・Z02・Z03・Z05・Z06・Z09・Z10・Z11・Z15・D00・D04・D05・D06・D09・D10・D12・D13・D14・D16・D17・D19・D20・D21・D23・D24・D25・D27・D31・D32・D33・D35・D38・D39・D42・D43・D46・D47
C00	Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・D01・D02・D03・D04・D06・D10・D13・D15・D16・D18・D19・D21・D24・D26・D27・D32・D34・D35・D39・D43・D47

なお、ZはL：ラッチ（フリップ・フロップ）の初期値を意味します。

c. データ置換

b.で得られた演算式に、a.よりD47～D31まではCRC16①の演算結果ですのでP7に記載した演算式を代入します。

例)最下位ビット（C0)での置き換え例
C0= Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・
D01・D02・D03・D04・D08・D10・D13・D15・D16・D18・D19・D21・D24・D26・D27・D32・D34・
D35・D39・D43・D47

P7より
D47 <= 　　C15= 　　X03・X04・X06・X09・X11・X12・X14・X15・
D00・D01・D03・D04・D06・D09・D11・D12・D17・D19・D20・D24・D28
D43 <= 　　C11= 　　X00・X02・X03・X04・X05・X06・X07・X08・X09・X10・X12・X13・X14・X15・
D00・D01・D02・D03・D05・D06・D07・D08・D09・D10・D11・D12・D13・D15・D16・
D17・D19・D20・D21・D23
D39 <= 　　C07= 　　X00・X01・X02・X03・X04・X05・X06・X08・X09・X10・X11・X12・X13・
D02・D03・D04・D05・D06・D07・D09・D10・D11・D12・D13・D14・D15・D16・D17・
D19・D20・D21・D23・D24・D25・D27
D35 <= 　　C03= 　　X00・X01・X02・X04・X05・X06・X07・X08・X09・X14・X15・
D00・D01・D06・D07・D08・D09・D10・D11・D13・D14・D15・D16・D17・D18・D19・
D20・D21・D23・D24・D25・D27・D28・D29・D31
D34 <= 　　C02= 　　X00・X01・X05・X07・X08・X09・X11・X12・X13・X15・
D00・D02・D03・D04・D06・D07・D08・D10・D14・D15・D16・D18・D21・D22・D25・
D26・D29・D30
D32 <= 　　C00= 　　X04・X05・X07・X10・X12・X13・X15・
D00・D02・D03・D05・D08・D10・D11・D16・D18・D19・D23・D27・D31

代入して
C0= Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・
D01・D02・D03・D04・D08・D10・D13・D15・D16・D18・D19・D21・D24・D26・D27・
X03・X04・X06・X09・X11・X12・X14・X15・
D00・D01・D03・D04・D06・D09・D11・D12・D17・D19・D20・D24・D28・
X00・X02・X03・X04・X05・X06・X07・X08・X09・X10・X12・X13・X14・X15・
D00・D01・D02・D03・D05・D06・D07・D08・D09・D10・D11・D12・D13・D15・D16・D17・
D19・D20・D21・D23・
X00・X01・X02・X04・X05・X06・X07・X08・X09・X14・X15・
D00・D01・D06・D07・D08・D09・D10・D11・D13・D14・D15・D16・D17・D18・D19・
D20・D21・D23・D24・D25・D27・D28・D29・D31・
X00・X01・X05・X07・X08・X09・X11・X12・X13・X15・
D00・D02・D03・D04・D06・D07・D08・D10・D14・D15・D16・D18・D21・D22・D25・
D26・D29・D30・
X04・X05・X07・X10・X12・X13・X15・
D00・D02・D03・D05・D08・D10・D11・D16・D18・D19・D23・D27・D31

同一項を削除して
C0= Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・
X01・X02・X03・X04・X05・X09・X10・X11・X12・X14・X15・
D00・D02・D05・D11・D12・D14・D15・D21・D22・D25・D30

d. 演算式

以上を全ての演算式において行うことで、CRC16演算器②′ を得ることが可能となります。

	演算式　（・：ExcuisiveOR）
C15	Z01・Z04・Z08・Z10・Z11・Z12・Z13・ X00・X02・X03・X04・X05・X06・X10・X11・X12・X13・X15・ D00・D07・D09・D10・D12・D13・D14・D15・D16・D18・D19・D20・D21・D22・D23・D24・D26・D27・D28・ D29・D31
C14	Z00・Z03・Z07・Z09・Z10・Z11・Z12・ X01・X03・X04・X05・X06・X07・X11・X12・X13・X14・ D01・D02・D05・D06・D09・D10・D11・D14・D15・D16・D18・D21・D22・D24・D25・D26・D29・ D30
C13	Z02・Z06・Z08・Z09・Z10・Z11・Z15・ X00・X02・X06・X08・X10・X14・ D00・D01・D04・D06・D15・D16・D19・D27・D30・D31
C12	Z01・Z05・Z07・Z08・Z09・Z10・Z14・ X01・X03・X07・X09・X11・X15・ D00・D01・D04・D05・D07・D10・D12・D19・D21・D22・D25・D31
C11	Z00・Z01・Z06・Z07・Z09・Z10・Z11・Z12・Z15・ X00・X03・X04・X06・X07・X08・X10・X11・X12・ D00・D06・D07・D11・D12・D14・D16・D18・D19・D20・D23・D27・D28・D29
C10	Z00・Z05・Z06・Z08・Z09・Z10・Z11・Z14・Z15・ X01・X04・X05・X07・X08・X09・X11・X12・X13・ D00・D01・D02・D03・D05・D08・D09・D11・D14・D15・D17・D18・D21・D22・D26・D27・D28・D29
C09	Z04・Z05・Z07・Z08・Z09・Z10・Z13・Z14・Z15・ X00・X02・X05・X06・X08・X09・X10・X12・X13・X14・ D00・D03・D08・D09・D11・D13・D15・D16・D17・D21・D22・D25・D26・D27・D28・D30・D31
C08	Z03・Z04・Z06・Z07・Z08・Z09・Z12・Z13・Z14・ X01・X03・X06・X07・X09・X10・X11・X13・X14・X15・ D00・D03・D04・D05・D07・D11・D14・D20・D23・D24・D25・D26・D27・D29・D30・D31
C07	Z02・Z03・Z05・Z06・Z07・Z08・Z11・Z12・Z13・ X02・X05・X08・X11・X13・X14・ D01・D03・D08・D09・D12・D17・D18・D25・D26・D27・D28・D29・D30
C06	Z01・Z02・Z04・Z05・Z06・Z07・Z10・Z11・Z12・ X03・X04・X05・X06・X07・X09・X10・X13・X14・ D01・D02・D03・D04・D06・D12・D13・D14・D19・D24・D26・D28・D29
C05	Z00・Z01・Z03・Z04・Z05・Z06・Z09・Z10・Z11・ X06・X08・X11・X12・X13・X14・ D01・D02・D03・D05・D06・D07・D10・D11・D12・D14・D15・D16・D18・D24・D25・D26・D28
C04	Z00・Z02・Z03・Z04・Z05・Z08・Z09・Z10・Z15・ X00・X07・X09・X12・X13・X14・X15・ D01・D02・D03・D05・D07・D08・D10・D11・D12・D13・D16・D17・D19・D20・D23・D24・D31
C03	Z01・Z02・Z03・Z04・Z07・Z08・Z09・Z14・ X01・X06・X10・X13・X14・X15・ D00・D02・D05・D06・D08・D11・D12・D13・D17・D18・D19・D20・D21・D22・D23・D26・D28・D30・D31
C02	Z00・Z02・Z03・Z04・Z06・Z07・Z10・Z11・Z12・ X03・X06・X07・X09・X14・X15・ D00・D01・D03・D04・D05・D06・D11・D13・D15・D19・D21・D22・D24・D25・D26・D28・D29・D30
C01	Z01・Z02・Z03・Z05・Z06・Z09・Z10・Z11・Z15・ X00・X05・X08・X12・X13・ D00・D02・D03・D04・D05・D06・D07・D09・D12・D13・D14・D15・D18・D19・D21・D22・D25・D27・D28・ D30
C00	Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・ X01・X02・X03・X04・X05・X09・X10・X11・X12・X14・X15・ D00・D02・D05・D11・D12・D14・D15・D21・D22・D25・D30

なお、XはH：ラッチ（フリップ・フロップ）、ZはL：ラッチ（フリップ・フロップ）の初期値を意味します。

③U：CRC16演算器③′演算式算出

a. データ幅64ビット

CRC16演算器③′では入力データ（32ビット）と1つ前の演算結果（16ビット）を用いて算出します。従来方法ではこの入力データの最後（次）にCRC16符号ビット（CRC16演算器②の結果：16ビット）を付加してCRC16の符号ビットを得ています。本発明で付加したS：CRC16演算器③′では入力データと同時にCRC16演算器①の結果（最終結果の2つ前）およびCRC16演算器②′の結果（最終結果の1つ前）を入力します。つまり入力データを64ビットとして演算式を算出します。この時、本来の入力データを下位ビット、CRC演算結果を上位ビット（下記参照）で入力します。

| 63　　　　　　　4847　　　　　3231　　　　　　　　　　　　0 |
| CRC16②演算結果 | CRC16①演算結果 | 入力データ |
| 16ビット | 16ビット | 32ビット |
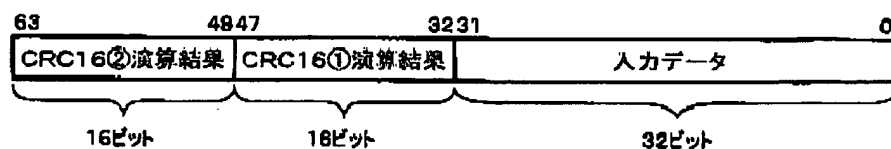
b. 演算式算出 — I

まず、データ幅64ビットでのCRC16演算式を算出します。これは先般お渡しした資料P5記載のCRC16生成回路から、64シフトした際の各FF出力となります。

	演算式 （・:ExculsiveOR）
C15	R01・R03・R04・R07・R08・R10・R11・R12・R13・R15・D00・D02・D03・D04・D05・D07・D08・D11・D12・D14・D18・D19・D20・D21・D23・D27・D30・D32・D33・D35・D36・D38・D41・D43・D44・D49・D51・D52・D58・D60
C14	R00・R02・R03・R06・R07・R09・R10・R11・R12・R14・R15・D00・D01・D03・D04・D05・D06・D08・D09・D12・D13・D15・D18・D20・D21・D22・D24・D28・D31・D33・D34・D36・D37・D39・D42・D44・D45・D50・D52・D53・D57・D61
C13	R01・R02・R05・R06・R08・R09・R10・R11・R13・R14・R15・D00・D01・D02・D04・D05・D06・D07・D09・D10・D13・D14・D16・D20・D21・D22・D23・D25・D29・D32・D34・D35・D37・D38・D40・D43・D45・D46・D51・D53・D54・D58・D62
C12	R00・R01・R04・R05・R07・R08・R09・R10・R12・R13・R14・R15・D00・D01・D02・D03・D05・D06・D07・D08・D10・D11・D14・D15・D17・D21・D22・D23・D24・D26・D30・D33・D35・D36・D38・D39・D41・D44・D46・D47・D52・D54・D55・D59・D63
C11	R00・R01・R06・R09・R10・R14・D01・D05・D06・D09・D14・D15・D16・D19・D20・D21・D22・D24・D30・D31・D32・D33・D34・D35・D37・D38・D39・D40・D41・D42・D43・D44・D45・D47・D48・D49・D51・D52・D53・D55
C10	R00・R05・R08・R09・R13・D02・D06・D07・D10・D15・D16・D17・D20・D21・D22・D23・D25・D26・D31・D32・D33・D34・D35・D36・D38・D39・D40・D41・D42・D43・D44・D45・D46・D48・D49・D50・D52・D53・D54・D56
C09	R04・R07・R08・R12・R15・D00・D03・D07・D08・D11・D16・D17・D18・D21・D22・D23・D24・D26・D27・D32・D33・D34・D35・D36・D37・D39・D40・D41・D42・D43・D44・D45・D46・D47・D49・D50・D51・D53・D54・D55・D57
C08	R03・R06・R07・R11・R14・R15・D00・D01・D04・D08・D09・D12・D17・D18・D19・D22・D23・D24・D25・D27・D28・D33・D34・D35・D36・D37・D38・D40・D41・D42・D43・D44・D45・D46・D47・D48・D50・D51・D52・D54・D55・D56・D58
C07	R02・R05・R06・R10・R13・R14・R15・D00・D01・D02・D05・D09・D10・D13・D18・D19・D20・D23・D24・D25・D26・D28・D29・D34・D35・D36・D37・D38・D39・D41・D42・D43・D44・D45・D46・D47・D48・D49・D51・D52・D53・D55・D56・D57・D59
C06	R01・R04・R05・R09・R12・R13・R14・D01・D02・D03・D06・D10・D11・D14・D19・D20・D21・D24・D25・D26・D27・D29・D30・D35・D36・D37・D38・D39・D40・D42・D43・D44・D45・D46・D47・D48・D49・D50・D52・D53・D54・D56・D57・D58・D60
C05	R00・R03・R04・R08・R11・R12・R13・D02・D03・D04・D07・D11・D12・D15・D20・D21・D22・D25・D26・D27・D28・D30・D31・D36・D37・D38・D39・D40・D41・D43・D44・D45・D46・D47・D48・D49・D50・D51・D53・D54・D55・D57・D58・D59・D61
C04	R02・R03・R07・R10・R11・R12・R15・D00・D03・D04・D05・D08・D12・D13・D16・D21・D22・D23・D26・D27・D28・D29・D31・D32・D37・D38・D39・D40・D41・D42・D44・D45・D46・D47・D48・D49・D50・D51・D52・D54・D55・D56・D58・D59・D60・D62
C03	R01・R02・R06・R09・R10・R11・R14・D01・D04・D05・D06・D09・D13・D14・D17・D22・D23・D24・D27・D28・D29・D30・D32・D33・D38・D39・D40・D41・D42・D43・D45・D46・D47・D48・D49・D50・D51・D52・D53・D55・D56・D57・D59・D60・D61・D63
C02	R00・R03・R04・R05・R07・R09・R11・R12・D03・D04・D06・D08・D10・D11・D12・D15・D19・D20・D21・D24・D25・D27・D28・D29・D31・D32・D34・D35・D36・D38・D39・D40・D42・D46・D47・D48・D50・D53・D54・D57・D58・D61・D62
C01	R02・R03・R04・R06・R08・R10・R11・R15・D00・D04・D05・D07・D09・D11・D12・D13・D16・D20・D21・D22・D25・D26・D28・D29・D30・D32・D33・D35・D36・D37・D39・D40・D41・D43・D47・D48・D49・D51・D54・D55・D58・D59・D62・D63
C00	R02・R04・R05・R08・R09・R11・R12・R13・R14・D01・D02・D03・D04・D06・D07・D10・D11・D13・D17・D18・D19・D20・D22・D26・D29・D31・D32・D34・D35・D37・D40・D42・D43・D48・D50・D51・D55・D59・D63

なお、RはR:ラッチ（フリップ・フロップ）の初期値を意味します。

c. データ置換

b.で得られた演算式に、a.よりD63〜D48まではCRC16②'の演算結果ですのでP11に記載した演算式を代入、D47〜D32まではCRC16①の演算結果ですのでP7に記載した演算式を代入、します。

例）最下位ビット（C0）での置き換え例

$C0 = R02 \cdot R04 \cdot R05 \cdot R08 \cdot R09 \cdot R11 \cdot R12 \cdot R13 \cdot R14 \cdot$
$D01 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D10 \cdot D11 \cdot D13 \cdot D17 \cdot D18 \cdot D19 \cdot D20 \cdot D22 \cdot D26 \cdot D29 \cdot D31 \cdot$
$D32 \cdot D34 \cdot D35 \cdot D37 \cdot D40 \cdot D42 \cdot D43 \cdot D48 \cdot D50 \cdot D51 \cdot D55 \cdot D59 \cdot D63$

P11より

$D63 \Leftarrow C15 = Z01 \cdot Z04 \cdot Z08 \cdot Z10 \cdot Z11 \cdot Z12 \cdot Z13 \cdot$
$X00 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X10 \cdot X11 \cdot X12 \cdot X13 \cdot X15 \cdot$
$D00 \cdot D07 \cdot D09 \cdot D10 \cdot D12 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot D19 \cdot D20 \cdot D21 \cdot D22 \cdot D23 \cdot D24 \cdot$
$D26 \cdot D27 \cdot D28 \cdot D29 \cdot D31$

$D59 \Leftarrow C11 = Z00 \cdot Z01 \cdot Z06 \cdot Z07 \cdot Z09 \cdot Z10 \cdot Z11 \cdot Z12 \cdot Z15 \cdot$
$X00 \cdot X03 \cdot X04 \cdot X06 \cdot X07 \cdot X08 \cdot X10 \cdot X11 \cdot X12 \cdot$
$D00 \cdot D06 \cdot D07 \cdot D11 \cdot D12 \cdot D14 \cdot D16 \cdot D18 \cdot D19 \cdot D20 \cdot D23 \cdot D27 \cdot D28 \cdot D29$

$D55 \Leftarrow C07 = Z02 \cdot Z03 \cdot Z05 \cdot Z06 \cdot Z07 \cdot Z08 \cdot Z11 \cdot Z12 \cdot Z13 \cdot X02 \cdot X05 \cdot X08 \cdot X11 \cdot X13 \cdot X14 \cdot$
$D01 \cdot D03 \cdot D08 \cdot D09 \cdot D12 \cdot D17 \cdot D18 \cdot D25 \cdot D26 \cdot D27 \cdot D28 \cdot D29 \cdot D30$

$D51 \Leftarrow C03 = Z01 \cdot Z02 \cdot Z03 \cdot Z04 \cdot Z07 \cdot Z08 \cdot Z09 \cdot Z14 \cdot X01 \cdot X08 \cdot X10 \cdot X13 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D05 \cdot D06 \cdot D08 \cdot D11 \cdot D12 \cdot D13 \cdot D17 \cdot D18 \cdot D19 \cdot D20 \cdot D21 \cdot D22 \cdot D23 \cdot D26 \cdot$
$D28 \cdot D30 \cdot D31$

$D50 \Leftarrow C02 = Z00 \cdot Z02 \cdot Z03 \cdot Z04 \cdot Z06 \cdot Z07 \cdot Z10 \cdot Z11 \cdot Z12 \cdot X03 \cdot X06 \cdot X07 \cdot X09 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D03 \cdot D04 \cdot D05 \cdot D06 \cdot D11 \cdot D13 \cdot D15 \cdot D19 \cdot D21 \cdot D22 \cdot D24 \cdot D25 \cdot D26 \cdot D28 \cdot$
$D29 \cdot D30$

$D48 \Leftarrow C00 = Z00 \cdot Z02 \cdot Z05 \cdot Z09 \cdot Z11 \cdot Z12 \cdot Z13 \cdot Z14 \cdot$
$X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X09 \cdot X10 \cdot X11 \cdot X12 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D05 \cdot D11 \cdot D12 \cdot D14 \cdot D15 \cdot D21 \cdot D22 \cdot D25 \cdot D30$

P7より

$D43 \Leftarrow C11 = X00 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X10 \cdot X12 \cdot X13 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D02 \cdot D03 \cdot D05 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D15 \cdot D16 \cdot D17 \cdot$
$D19 \cdot D20 \cdot D21 \cdot D23$

$D42 \Leftarrow C10 = X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X11 \cdot X12 \cdot X13 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D14 \cdot D16 \cdot D17 \cdot$
$D18 \cdot D20 \cdot D21 \cdot D22 \cdot D24$

$D40 \Leftarrow C08 = X00 \cdot X01 \cdot X02 \cdot X03 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X09 \cdot X10 \cdot X11 \cdot X12 \cdot X13 \cdot X14 \cdot$
$D01 \cdot D02 \cdot D03 \cdot D04 \cdot D05 \cdot D06 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D12 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot$
$D19 \cdot D20 \cdot D22 \cdot D23 \cdot D24 \cdot D26$

$D37 \Leftarrow C05 = X00 \cdot X01 \cdot X02 \cdot X03 \cdot X04 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X10 \cdot X11 \cdot$
$D04 \cdot D05 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D11 \cdot D12 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D17 \cdot D18 \cdot D19 \cdot D21 \cdot$
$D22 \cdot D23 \cdot D25 \cdot D26 \cdot D27 \cdot D29$

$D35 \Leftarrow C03 = X00 \cdot X01 \cdot X02 \cdot X04 \cdot X05 \cdot X06 \cdot X07 \cdot X08 \cdot X09 \cdot X14 \cdot X15 \cdot$
$D00 \cdot D01 \cdot D06 \cdot D07 \cdot D08 \cdot D09 \cdot D10 \cdot D11 \cdot D13 \cdot D14 \cdot D15 \cdot D16 \cdot D17 \cdot D18 \cdot D19 \cdot D20 \cdot$
$D21 \cdot D23 \cdot D24 \cdot D25 \cdot D27 \cdot D28 \cdot D29 \cdot D31$

$D34 \Leftarrow C02 = X00 \cdot X01 \cdot X05 \cdot X07 \cdot X08 \cdot X09 \cdot X11 \cdot X12 \cdot X13 \cdot X15 \cdot$
$D00 \cdot D02 \cdot D03 \cdot D04 \cdot D06 \cdot D07 \cdot D08 \cdot D10 \cdot D14 \cdot D15 \cdot D16 \cdot D18 \cdot D21 \cdot D22 \cdot D25 \cdot D26 \cdot$
$D29 \cdot D30$

$D32 \Leftarrow C00 = X04 \cdot X05 \cdot X07 \cdot X10 \cdot X12 \cdot X13 \cdot X15$
$D00 \cdot D02 \cdot D03 \cdot D05 \cdot D08 \cdot D10 \cdot D11 \cdot D16 \cdot D18 \cdot D19 \cdot D23 \cdot D27 \cdot D31$

代入して
C0＝Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・

C0＝R02・R04・R05・R08・R09・R11・R12・R13・R14・
D01・D02・D03・D04・D06・D07・D10・D11・D13・D17・D18・D19・D20・D22・D26・D29・D31・
Z01・Z04・Z08・Z10・Z11・Z12・Z13・
X00・X02・X03・X04・X05・X06・X10・X11・X12・X13・X15・
D00・D07・D09・D10・D12・D13・D14・D15・D16・D18・D19・D20・D21・D22・D23・D24・
D26・D27・D28・D29・D31・
Z00・Z01・Z06・Z07・Z09・Z10・Z11・Z12・Z15・
X00・X03・X04・X06・X07・X08・X10・X11・X12・
D00・D06・D07・D11・D12・D14・D16・D18・D19・D20・D23・D27・D28・D29・
Z02・Z03・Z05・Z06・Z07・Z08・Z11・Z12・Z13・X02・X05・X08・X11・X13・X14・
D01・D03・D08・D09・D12・D17・D18・D25・D26・D27・D28・D29・D30・
Z01・Z02・Z03・Z04・Z07・Z08・Z09・Z14・X01・X08・X10・X13・X14・X15・
D00・D02・D05・D06・D08・D11・D12・D13・D17・D18・D19・D20・D21・D22・D23・D26・
D28・D30・D31・
Z00・Z02・Z03・Z04・Z06・Z07・Z10・Z11・Z12・X03・X06・X07・X09・X14・X15・
D00・D01・D03・D04・D05・D06・D11・D13・D15・D19・D21・D22・D24・D25・D26・D28・
D29・D30・
Z00・Z02・Z05・Z09・Z11・Z12・Z13・Z14・
X01・X02・X03・X04・X05・X09・X10・X11・X12・X14・X15・
D00・D02・D05・D11・D12・D14・D15・D21・D22・D25・D30・
X00・X02・X03・X04・X05・X06・X07・X08・X09・X10・X12・X13・X14・X15・
D00・D01・D02・D03・D05・D06・D07・D08・D09・D10・D11・D12・D13・D15・D16・D17・
D19・D20・D21・D23・
X01・X02・X03・X04・X05・X06・X07・X08・X09・X11・X12・X13・X14・X15・
D00・D01・D02・D03・D04・D06・D07・D08・D09・D10・D11・D12・D13・D14・D16・D17・
D18・D20・D21・D22・D24・
X00・X01・X02・X03・X04・X05・X06・X07・X09・X10・X11・X12・X13・X14・
D01・D02・D03・D04・D05・D06・D08・D09・D10・D11・D12・D13・D14・D15・D16・D18・
D19・D20・D22・D23・D24・D26・
X00・X01・X02・X03・X04・X06・X07・X08・X09・X10・X11・
D04・D05・D06・D07・D08・D09・D11・D12・D13・D14・D15・D16・D17・D18・D19・D21・
D22・D23・D25・D26・D27・D29・
X00・X01・X02・X04・X05・X06・X07・X08・X09・X14・X15・
D00・D01・D06・D07・D08・D09・D10・D11・D13・D14・D15・D16・D17・D18・D19・D20・
D21・D23・D24・D25・D27・D28・D29・D31・
X00・X01・X05・X07・X08・X09・X11・X12・X13・X15・
D00・D02・D03・D04・D06・D07・D08・D10・D14・D15・D16・D18・D21・D22・D25・D26・
D29・D30・
X04・X05・X07・X10・X12・X13・X15
D00・D02・D03・D05・D08・D10・D11・D16・D18・D19・D23・D27・D31

同一項を削除して
C0＝R02・R04・R05・R08・R09・R11・R12・R13・R14・
Z02・Z03・Z04・Z07・Z09・Z10・Z11・
X02・X04・X05・X06・X08・X12・X13・
D01・D05・D08・D09・D11・D12・D13・D16・D17・D18・D21・D22・D24・D30・D31

d. 演算式

以上を全ての演算式において行うことで、CRC16演算器③′を得ることが可能となります。

	演算式 （・：ExculsiveOR）
C15	R01・R03・R04・R07・R08・R10・R11・R12・R13・R15・Z03・Z04・Z05・Z08・Z10・Z11・Z12・X01・X05・X06・X07・X09・X14・D00・D01・D02・D06・D09・D16・D19・D20・D23・D24・D26・D28・・・
C14	R00・R02・R03・R06・R07・R09・R10・R11・R12・R14・R15・Z00・Z04・Z05・Z06・Z09・Z11・Z12・Z13・X00・X03・X05・X08・X09・X11・X12・X13・X14・D00・D03・D04・D05・D06・D07・D08・D11・D13・D15・D17・D18・D19・D24・D27・D28・D29・D30
C13	R01・R02・R05・R06・R08・R09・R10・R11・R13・R14・R15・Z00・Z01・Z02・Z06・Z07・Z09・Z10・Z11・X01・X04・X05・X06・X07・X08・X11・X12・X13・X15・D01・D03・D04・D09・D11・D14・D15・D18・D20・D21・D22・D23・D31
C12	R00・R01・R04・R05・R07・R08・R09・R10・R12・R13・R14・R15・Z00・Z01・Z02・Z03・Z04・Z05・Z08・Z11・Z13・Z15・X00・X03・X04・X05・X06・X07・X10・X11・X12・X14・D06・D09・D10・D11・D13・D15・D21・D22・D24・D26・D31
C11	R00・R01・R06・R09・R10・R14・Z00・Z01・Z04・Z06・Z07・Z08・Z15・X03・X04・X05・X06・X07・X11・X13・X14・X15・D02・D04・D05・D06・D07・D09・D10・D12・D16・D20・D21・D22・D23・D24・D25・D26・D28・D31
C10	R00・R05・R08・R09・R13・Z01・Z02・Z04・Z08・Z09・Z10・Z12・Z13・Z15・X02・X05・X09・X10・X11・X13・D00・D02・D03・D04・D06・D11・D14・D15・D16・D18・D19・D21・D22・D24・D26・D28・D31
C09	R04・R07・R08・R12・R15・Z02・Z03・Z05・Z09・Z10・Z11・Z13・Z14・X01・X04・X08・X09・X10・X12・D00・D01・D02・D04・D08・D10・D12・D13・D14・D16・D17・D19・D25・D26・D27・D29・D30
C08	R03・R06・R07・R11・R14・R15・Z03・Z04・Z06・Z10・Z11・Z12・Z14・Z15・X00・X04・X06・X07・X08・X12・X14・X15・D00・D01・D05・D07・D09・D15・D18・D21・D22・D24・D25・D26・D27・D29・D30
C07	R02・R05・R06・R10・R13・R14・R15・Z00・Z02・Z05・Z09・Z10・Z12・Z13・Z14・X01・X02・X03・X07・X10・X11・X13・X14・D00・D01・D02・D03・D04・D05・D06・D08・D09・D12・D13・D14・D15・D17・D19・D26・D28・D30
C06	R01・R04・R05・R09・R12・R13・R14・Z00・Z01・Z02・Z03・Z04・Z06・Z07・Z09・Z13・X00・X03・X04・X05・X06・X11・X13・X14・X15・D00・D02・D03・D04・D08・D11・D13・D16・D17・D20・D21・D22・D23・D24・D25
C05	R00・R03・R04・R08・R11・R12・R13・Z00・Z01・Z03・Z04・Z07・Z08・Z09・Z10・Z11・Z12・Z13・X01・X09・X11・X13・X15・D00・D02・D04・D05・D08・D17・D20・D23・D26・D27・D28・D29・D31
C04	R02・R03・R07・R10・R11・R12・R15・Z01・Z02・Z07・Z08・Z09・Z11・Z14・Z15・X00・X03・X04・X06・X08・X09・X10・X11・D01・D03・D04・D09・D11・D14・D15・D18・D20・D21・D22・D23・D31
C03	R01・R02・R06・R09・R10・R11・R14・Z00・Z02・Z03・Z04・Z05・Z07・Z08・Z09・Z13・X02・X04・X05・X06・X07・X08・X10・X11・X12・X14・X15・D00・D02・D03・D12・D13・D14・D15・D16・D24・D26・D27・D28・D29・D30・D31
C02	R00・R03・R04・R05・R07・R09・R11・R12・Z00・Z01・Z02・Z04・Z06・Z07・Z10・Z12・Z15・X01・X02・X05・X09・X12・X13・X14・D00・D01・D02・D03・D04・D05・D09・D12・D18・D20・D23・D25・D30・D31
C01	R02・R03・R04・R06・R08・R10・R11・R15・Z01・Z03・Z07・Z08・Z09・Z12・Z14・X00・X02・X03・X05・X08・X09・X10・X13・X14・X15・D02・D03・D04・D07・D08・D09・D10・D11・D12・D14・D15・D17・D18・D19・D25・D26・D27・D28・D29・D31・
C00	R02・R04・R05・R08・R09・R11・R12・R13・R14・Z02・Z03・Z04・Z07・Z09・Z10・Z11・X02・X04・X05・X06・X08・X12・X13・D01・D05・D08・D09・D11・D12・D13・D16・D17・D18・D21・D22・D24・D30・D31

なお、RはR：ラッチ（フリップ・フロップ）、XはH：ラッチ（フリップ・フロップ）、ZはL：ラッチ（フリップ・フロップ）の初期値を意味します。
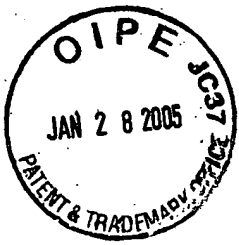
PATENT APPLICATION

Appln. Of: KOTAKA
Filed: March 4, 2002
For: ARITHMETIC OPERATION METHOD FOR CYCLIC...
Docket: NEC N01321

Received:

jc971 U.S. PTO
10/090302
03/04/02

1. Form PTO-2038 (1 pg)
2. Transmittal letter (1 pg)
5. Declaration/Power of Attorney (3 pgs)
3. Spec (49 pgs) Claims (7 pgs) Abstract (1 pg)
4. Drawings (20 pgs)
5. Assignment w/cover sheet (3 pgs)
6. Change of Attorney Address w/Cert. Of
   Express Mail (2 pgs)

03/04/02 SMM